Projekt "Autonomer Schnelltransporter"

Dokumentation PREN 2

Team 31

Philipp Burch
Remo Frank
Adrian Marti
Ramon Rohner
Matthias Rüegg
Sven Ulrich
Jean-Marc Vogel

Horw, 17. Juni 2012

Management Summary

Die Ausgangslage dieser Arbeit bildet die Aufgabenstellung des Moduls TA.BA_PREN1.H1101. Diese Arbeit stellt die Fortsetzung des Moduls dar. Es wird die Realisation des „autonomen Schnelltransporters“, der auf einem vorgegebenen Parcours einen Würfel aufhebt und ins Ziel befördert, aufgezeigt.

Neben der eigentlichen Umsetzung des Konzeptes aus PREN1 wird in dieser Arbeit auch der Bezug zum evaluierten Zielmarkt der Industrieputzroboter beleuchtet. Es wurde ersichtlich, dass mit der Marktanwendung die Marktführerschaft im Bereich der autonomen Putzroboter angestrebt werden kann. Dies nicht zuletzt aufgrund der innovativen Orientierung mittels Tiefenbild und der hohen Nachfrage nach effizienten Reinigungslösungen.

Die Änderungen und Abweichungen zum Parcours-Roboter sind vor allem die Grösse und die Würfelaufnahme. Während im Parcours noch auf einen Greifer gesetzt wird, muss das marktfähige Fahrzeug mit Bürsten (Kehrmaschine) oder mit Scheuertellern und Sauger (Scheuersaugmaschine) ausgestattet sein. Zudem kommt die Implementierung von drei Reinigungsprogrammen, welche mit der 3D Raumerkennung einhergehen müssen.

Zum Softwareteil gehören die Module der Robotersteuerung und des Emulators. Dazu gehört auch die Bewegungssteuerung mittels eines virtuellen Liniensensors, die Verarbeitung des Tiefenbilds der Kinect und die Modellierung des kompletten Roboters in einem Emulator zur Vereinfachung der Softwareentwicklung.

Die Elektronik des Roboters umfasst einerseits den Single-Board-Computer TS-7500 und andererseits die Hauptplatine. Neben dem Eingangsteil mit Unterspannungsabschaltung und dem DC-Motortreiber sind auch verschiedene Schnittstellen zum Anschluss von Servos und Tastern vorhanden. Zu Überwachungszwecken sind ein RS232-Interface und ein ADC zur Spannungs- und Strommessung eingebaut, um die Tiefentladung des Akkus zu verhindern.

Der mechanische Aufbau ist als zweirädriges Fahrzeug mit einer Kugelrolle als vorderer Abstützung ausgeführt. Die beiden Räder kombinieren die Navigation als Panzersteuerung mit dem eigentlichen Antrieb bei einer Maximalgeschwindigkeit von über 2m/s. Zur Ergreifung des Würfels dient ein Zangengreifer an der Front des Roboters. Besonderes Augenmerk wurde auch auf die Befestigung der Kinect-Kamera am hinteren Teil des Fahrzeugs gelegt.

Inhalt

Management Summary 2

Abbildungsverzeichnis 5

Abkürzungsverzeichnis 7

1. Einführung 8

2. A-Markt 9

2.1. Dominanzmatrix 10
2.2. Auswertung der Dominanzmatrix 11
2.3. Marktsegmentierung 11
2.4. Mögliche Einsatzgebiete 13
2.5. Konkurrenzanalyse 13
2.6. SWOT-Analyse 14
2.7. Wettbewerbsvorteile 15
2.8. Skalen- und Mengeneffekt 16

3. Produktanforderungen 17

3.1. Einsatzbereich/Umgebung 17
3.2. Beispielumgebungen und Programme 18

4. Abweichungen Prototyp / marktfähiges Produkt 20

4.1. Produktion / Verkauf 20
4.2. Fahrzeugeigenschaften und Funktionen 21
4.3. Bedienung / Einstellbarkeit 22
4.4. Erweiterbarkeit 22

5. Designstudie 23

5.1. Anforderungen an das Design 23
5.2. Designanalyse 23
5.3. Bestehende Modelle von Scheuersaug- und Kehrmaschinen 24
5.4. Entwürfe für PREN 25
5.5. Entwürfe für Markt 26
5.6. Beispiele vom Markt 27

6. Beschreibung des Prototyps 29

6.1. Übersicht 29
6.2. Parcours-Ablauf 31
7. Software 32
7.1. Komponenten 32
7.2. Bewegungssteuerung 35
7.3. Programmablauf 39
7.4. Bildverarbeitung 42
7.5. Emulator 46
8. Elektronik 48
8.1. Blockschaltbild der gesamten Elektronik 48
8.2. Anschlüsse der Hauptplatine 49
8.3. Funktion der Schaltung 50
8.4. Spannungsregler 51
8.5. Motorentreiber 53
8.6. Encoder 54
8.7. Strommessung der Servos 55
8.8. Optionale Baugruppen 56
8.9. EMV-Massnahmen 56
8.10. FPGA 57
9. Mechanik 59
9.1. Übersicht 59
9.2. Antrieb und Fahrgestell 60
9.3. Greif- und Hebemechanismus 61
9.4. Kinect- und Verschalungsbefestigung 63
9.5. Optimierungsvorschläge für Serienmodell 64
10. Bedienungsanleitung 65
10.1. Wichtige Sicherheitsanweisungen 65
10.2. Leistungsmerkmale 65
10.3. Inbetriebnahme 66
10.4. Starten des Programms 66
10.5. Problembehandlung 73
11. Diskussion 75

11.1. Zeitlicher Aufwand 75
11.2. Entwicklungskosten 76
11.3. Lessons Learned 77
12. Testprotokoll 78
13. Literaturverzeichnis 80

Abbildungsverzeichnis

Abbildung 1: Bisherige Putzroboter: Kehrmaschine Wetrok 11
Abbildung 2: Hako Scheuersaugmaschine 13
Abbildung 3: Dachrinnen-Reiniger 14
Abbildung 4: Erfahrungskurve 16
Abbildung 7: Lagerhalle (123rf.com) 18
Abbildung 5: Raum-erkennung mit Route (aktion.kobold.vorwerk.com) 18
Abbildung 6: Grundriss Lagerhalle 18
Abbildung 8: Grundriss Produktionshalle 19
Abbildung 9: Produktionshalle (theaustin.com) 19
Abbildung 10: Grundriss Werkstatt 19
Abbildung 11: Werkstatt (metallbau-boehringer.de) 19
Abbildung 12: Scheuersaugmaschine (reinigungsberater.de) 20
Abbildung 13: x-box 360 mit Kinect 23
Abbildung 14: Neato und Roomba von iRobot (www.blogcdn.com) 23
Abbildung 15: Wetrok Kehrmaschine (wetrok.ch) 24
Abbildung 16: Kärcher Scheuersaugmaschine (kärcher.ch) 24
Abbildung 17: Entwurf Gehäuse für PREN 25
Abbildung 18: Entwurf Fahrzeug für PREN 25
Abbildung 19: Fahrzeug von der Seite und von vorne 25
Abbildung 20: Entwurf für Marktanwendung gross 26
Abbildung 21: Entwurf für Marktanwendung klein 26
Abbildung 22: Kehrprinzip (www.wetrok.ch) 26
Abbildung 23: Kehrmaschine von Haaga (industriebedarf-staplerteam.de) 27
Abbildung 24: Scheuersaugmaschine von Tennant (directindustry.de) 27
Abbildung 25: Concept Car (nzz.ch) 28
Abbildung 26: Übersicht Funktionsmuster 29
Abbildung 27: Ablauf des Parcours 31
Abbildung 28: Grobe Sicht auf die Softwarearchitektur 32
Abbildung 29: Klassendiagramm der Steuerungssoftware 33
Abbildung 30: Blockschema des Regelkreises 35

Abbildung 31: Virtuelle Linie im Emulator 36
Abbildung 32: Richtungsbestimmung 36
Abbildung 33: Detaillierter Parcours-Ablauf 40
Abbildung 34: Blockdiagramm Systemaufbau 40
Abbildung 35: Zusammenhand der Koordinatensysteme....42
Abbildung 36: Connected Component Labeling....44
Abbildung 37: Kinect-Tiefenbild mit Debug-Infos 45
Abbildung 38: Screenshot Emulator 46
Abbildung 39: Blockdiagramm des Roboters in Simulink 47
Abbildung 40: Blockdiagramm Systemaufbau 47
Abbildung 41: Blockschema der Elektronik....48
Abbildung 42: Unterspannungs-Abschaltung (undervoltage lockout) 50
Abbildung 43: 5V-Speisung für den SBC 51
Abbildung 44: 12V-Speisung für die Logik des DRV8412 52
Abbildung 45: Beschalteter Motorentreiben DRV8412....53
Abbildung 46: Encoderbaustein AS5040 54
Abbildung 47: Strommessung eines Servos 55
Abbildung 48: Ansteuerung des Piezo-Summers....56
Abbildung 49: FPGA-Übersicht....57
Abbildung 50: Getriebe....60
Abbildung 54: FEM-Simulation 61
Abbildung 51: Ansicht von Oben 61
Abbildung 52: Ansicht von Unten....61
Abbildung 53: Seitenansicht 61
Abbildung 56: Verlängerter und gefärbter Greifer 62
Abbildung 55: Greifarm....62
Abbildung 57: Fuss der Kinectkamera 63
Abbildung 58: Kinectbefestigung von unten....63

Abkürzungsverzeichnis

F	Festforderung
M	Mindestforderung
W	Wunsch
P	Parcours-Leiter (für PREN: Modulleitung, für Produktivbereich: Benutzer)
T	Teammitglieder
S	Teammitglieder Software (jv,pb,rf)
H	Teammitglieder Hardware (pb,rf,am,rr,su)
W	Teammitglieder Wirtschaftsingenieur | Innovation (mr,rr)
pb	Philipp Burch
rf	Remo Frank
am	Adrian Marti
rr	Ramon Rohner
mr	Matthias Rüegg
su	Sven Ulrich
jv	Jean-Marc Vogel
ADC	Analog to digitial converter, Analog-digital-Wandler
ARM	Advance RISC machine, eine Prozessorarchitektur
CPU	Central processing unit, zentrale Recheneinheit (Mikroprozessor)
DDS	Direct digital synthesis, ein Verfahren zur digitalen Erzeugung von Signalverläufen
EMV	Elektromagnetische Verträglichkeit
FPGA	Field programmable gate array, programmierbarer Logikbaustein
HAL	Hardware abstraction layer, Abstrahierung der Hardware
IC	Integratec circuit, Integrierter Schaltkreis
LED	Light emitting diode, Leuchtdiode
PWM	Pulse width modulation, Pulsbreitenmodulation
RTC	Realtime clock, Echtzeituhr (mit Stützbatterie)
SBC	Single board computer, Einplatinencomputer

1. Einführung

Die Ausganslage bildet die Aufgabenstellung aus dem Projektmodul Produktentwicklung 1 (PREN1). Das Konzept für den autonomen Schnelltransporter, welcher auf einem vorgegebenen Parcours einen Würfel aufladen und möglichst schnell ins Ziel bringen soll, wurde im vergangenen Herbstsemester in PREN1 erstellt. Nun geht es in PREN2 um die Realisierung des Konzeptes, welche diese Arbeit dokumentiert.

Um diese exemplarische Aufgabenstellung zu lösen, wurde das Know-how aus allen Disziplinen benötigt. Um eine Übersicht über die geplante Marktanwendung zu erhalten, beginnt die Arbeit mit dem Marktteil. Bei den Anforderungen an das Produkt werden die potentiellen Einsatzbereiche, Beispielprogramme und Umgebungen spezifiziert. In der Designstudie werden bestehende Produkte analysiert und anschliessend Entwürfe für die PREN- sowie Marktanwendung abgebildet. Den Kern der Arbeit bildet die Umsetzung der einzelnen Disziplinen (Software, Elektronik, Mechanik und Design, Kapitel 6 bis 9). Dort wird genauer auf die realisierte Lösung eingegangen und die verschiedenen Komponenten der Teillösungen beschrieben.

Kapitel 10 ist die Bedienungsanleitung für das Funktionsmuster.

In der Diskussion werden abschliessend der zeitliche Aufwand, Entwicklungskosten und Erfahrungen beschrieben.

Den Schluss bildet das Testprotokoll und das Literaturverzeichnis.

2. A-Markt

Im Gesamtkonzept des Moduls TA.BA_PREN1.H1101 wurde im Kapitel 3 Markt eine Umfeldanalyse durchgeführt. Basierend auf der Umfeldanalyse wurden die lukrativen Märkte eruiert. Diese wurden in der Dominanzmatrix untereinander verglichen. Die Kriterien bildeten die Zugänglichkeit, sowie die Abweichungen der Funktion des Roboters mit den Vorgaben des Wettbewerbs. Die Dominanzmatrix ist auf nachfolgender Seite abgebildet. Der Industrieputzroboter schnitt am besten ab. In diesem Segment liegt das grösste Potential. Die Abweichungen scheinen gering, die Zugänglichkeit gut und Konkurrenzprodukte sind wenig vorhanden. Auch gibt es viele potentielle Abnehmer. Der A-Markt wurde somit anhand der Situationsanalyse in PREN1 und der Dominanzmatrix als das Segment der (autonomen) Industrieputzroboter festgelegt. Der Name Industrieputzroboter wurde so gewählt, weil sofort ersichtlich ist, dass der Absatz auf business-to-business (B2B) Ebene erfolgt. Dies weil sich die Anschaffung erst ab einer bestimmten Grösse lohnt. Der Absatz beschränkt sich aber nicht nur auf die produzierende Industrie, auch gehören allgemein Betriebe mit Räumlichkeiten dazu, die entsprechender Reinigung bedürfen. Die genauere Segmentierung erfolgt nach der Dominanzmatrix. Anschliessend wurden die Stärken, Schwächen, Chancen und Risiken in der SWOT-Analyse analysiert. Die Beschreibung der Wettbewerbsvorteile bildet den Schluss.

2.1. Dominanzmatrix

Legende

Eingabefeld

> 0 besser

= 0 gleich gut

< 0 schlechter

Betrag Lösungsabstand

a: Haushaltsputzroboter

b: Industrieputzroboter

c: Strassenputzer

d: Touristenführer

e: Obst- oder Golfballsammler

f: Aufklärungs- und Bergungsroboter

g: Überwachungsroboter

Datum

10.11.11

		1	2	3	4	5	6	7
Lösung		a	b	c	d	e	f	g
im Vergleich zu	a		1	-1	-2	0	1	0
	b	-1		-1	-2	0	-1	0
	c	0	1		-1	0	1	0
	d	2	2	1		1	1	0
	e	0	0	0	-1		1	0
	f	-1	1	-1	-1	-1		-1
	g	0	0	0	0	0	1	
Summe		0	5	-2	-7	0	4	-1
Rangfolge		3	1	5	6	3	2	4

Rangfolge	
1	Industrieputzroboter
2	Aufklärungs- und Bergungsroboter
3	Obst- oder Golfballsammler
4	Haushaltsputzroboter
5	Überwachungsroboter
6	Strassenputzer
7	Touristenführer

2.2. Auswertung der Dominanzmatrix

In einer Dominanzmatrix werden die verschiedenen Varianten miteinander verglichen, um herauszufinden welches die optimale Lösung wäre. Hier haben wir die Einsatzgebiete auf ihre Marktzugänglichkeit und ob die Funktionen mit denen des Parcours übereinstimmen berücksichtigt.

Der Industrieputzroboter hat am besten abgeschnitten. Wir sehen dort das grösste Potential. Die nötigen Funktionen sind auch mit denen des Parcours vergleichbar.

Als zweitbestes hat der Aufklärungs- und Bergungsroboter abgeschnitten. Dieser Roboter müsste aber nicht autonom sein, da er stets überwacht wird.

Beim Obstsammler sehen wir kein grosses Marktpotential. Er wäre allerdings einfach zu modifizieren und als Golfballsammler einzusetzen.

Für den Haushaltsputzroboter gibt es einen grossen Markt, aber auch eine grosse Konkurrenz. Aus diesem Grund hat dieses Produkt nicht gut abgeschnitten.

Beim Strassenputz-, Überwachungs- und Touristenführer-Roboter sehen wir keine grosse Verbindung zwischen den Parcours Anforderungen in PREN und dem, was er später können muss. Aus diesem Grund haben diese nicht so gut abgeschnitten. Den Strassenputzer kann man nicht alleine arbeiten lassen. Zudem scheint der Markt nicht lukrativ.

2.3. Marktsegmentierung

Es wird eine Basisversion des Roboters geben um den Massenmarkt zu erreichen. Zudem besteht die Möglichkeit den Roboter zu modifizieren um auch im Premium-Markt Fuss zu fassen.

In der Schweiz hat es über 300'000 KMUs. Von diesen sind aber ca. 270'000 Kleinst-Unternehmen (bis 9 Mitarbeiter). Ein Kleinstunternehmen käme kaum als Abnehmer in Frage. Auch bei kleinen Unternehmen (10-49 Mitarbeiter) wird es schwierig. Das grosse Potential liegt bei den mittleren und Grossunternehmen.

Abbildung 1: Bisherige Putzroboter: Kehrmaschine Wetrok

Die meisten Grossunternehmen sind nicht in der Industrie oder im Handel tätig und brauchen somit auch keinen Industrieputzroboter. Schaut man aber Detailhändler wie

Migros, Coop usw. an, so zählen diese nur als ein Unternehmen. Jeder dieser Unternehmen hat aber hunderte Filialen in der Schweiz. Somit könnten auch grössere Mengen an denselben Anbieter verkauft werden. Um diese Abnehmer anzusprechen ist die Basisversion zu einem günstigen Preis erhältlich.

Mittlere Unternehmen, welche in der Industrie tätig sind, setzen vermehrt auf Qualität und individuelle Lösungen. Deshalb werden Upgrades auf die Basisversionen angeboten. Diese könnten aus einem zusätzlichen Hebelarm, 360° schwenkbarer Kamera, extra langer Laufzeit, Garantieleistungen usw. bestehen. Der Roboter kann beispielsweise nach Feierabend die Werkstätten wischen. In der Forschung könnte er als Transporter in sterilen Räumen oder Reinräumen eingesetzt werden.

Grössenklassen	**Unternehmen**		**Beschäftigte**	
nach Vollzeitäquivalenten	**Anzahl**	**%**	**Anzahl**	**%**
KMU (bis 249)	**311'707**	**99.6**	**2'327'802**	**66.6**
Mikrounternehmen (bis 9)	272'346	87.1	869'206	24.9
Kleine Unternehmen (10-49)	33'183	10.6	760'780	21.8
Mittlere Unternehmen (50-249)	6'178	2.0	697'816	20.0
Grosse Unternehmen (250 und mehr)	**1'154**	**0.4**	**1'166'269**	**33.4**
Total	**312'861**	**100.0**	**3'494'071**	**100.0**

Tabelle 1: Marktwirtschaftliche Unternehmen und Beschäftigte nach Grössenklassen, 2008

2.4. Mögliche Einsatzgebiete

Industrie	Kehr-maschine	Scheuer-sauger
Metallindustrie	✓	
Holzindustrie	✓	
Recycling und Abfallindustrie	✓	
Chemische Industrie (Zement, Glas, Kunststoff, Papier)	✓	
Konsumgüterindustrie (Möbel, Textil, Spielwaren)	✓	✓
Hallenreinigung		
Lagerhallen (Handel und Logistik)	✓	✓
Messehallen	✓	✓
Turnhallen		✓
Detailhandel		
Verbrauchsgüter	✓	✓
Gebrauchsgüter	✓	✓

Die Lebensmittelindustrie wird vorerst nicht beachtet, da es dort zu viele und strenge Regelungen im Bereich der Hygiene gibt. Auch der Gesundheitsbereich wie Krankenhäuser, Pharmaindustrie usw. wird aufgrund der strengen Richtlinien als Absatzsegment nicht angestrebt.

2.5. Konkurrenzanalyse

Grösste Konkurrenten im B2B-Bereich für Reinigungsmaschinen sind Wetrok, Kärcher und Hako. Wobei Hako auf grössere Maschinen spezialisiert ist. Weiter gibt es noch diverse kleinere und weniger bekannte Anbieter, wie cleanfix, Nilfisk, Nilco, Oertzen, Steiners usw. Keiner der erwähnten Anbieter hat eine autonome Reinigungsmaschine im Angebot.

Abbildung 2: Hako Scheuersaugmaschine

Der grösste Konkurrent im autonomen Bereich ist iRobot. iRobot ist der grösste Hersteller von autonomen Staubsaugern für den Haushalt. Zusätzlich bieten sie weitere Reinigungsroboter für private Haushalte an, wie Poolreiniger oder Dachrinnenreiniger. Ausserdem stellen sie Roboter für Militär, Unterwassereinsätze und Forschung her. Das Technische Know-how um auch selbständige Reinigungsmaschinen für die Industrie herzustellen wäre vorhanden.

Abbildung 3: Dachrinnen-Reiniger

2.6. SWOT-Analyse

Die SWOT-Analyse zeigt die Stärken, Schwächen, Chancen und Risiken auf. Die Stärken und Schwächen resultieren aus der internen Analyse, die Chancen und Risiken aus der Umweltanalyse (extern). In der Matrix wird jedem externen Faktor ein interner Faktor gegenübergestellt. Nachfolgend ist die SWOT-Matrix abgebildet.

SWOT **Industrieputzroboter**	**Stärken: (S)** - Interdisziplinäres Projektteam - Modifizierbar	**Schwächen: (W)** - Bedrohung von Arbeitsplätzen - Budget von CHF 600.-
Chancen: (O) - Erster solcher Roboter - Effizienz - Hohes Wachstum - Akzeptanz - Hohes Potential in der Logistik - Leichte Marktzugänglichkeit - Innovation durch einzigartige Orientierung (Tiefenbild)	**SO-Strategien (Investieren)** - Marktführerschaft für Lagerreinigungs-Roboter - Neue/Verbesserte Orientierungstechnologien - Effizienteres Putzen	**WO-Strategien (ausgleichen)** - Roboter für Arbeiten die keiner gern macht - Unterstützung der bisherigen Reinigungskräfte durch Roboter - Mehrere Roboter aufgrund tiefer Kosten einsetzbar
Risiken: (T) - Neue Wettbewerber - Steigende Ressourcenkosten - Strengere Gesetzgebung Politik (Subventionen)	**ST-Strategien (Absichern)** - Innovativ und vorausschauend - Umbrüche frühzeitig erkennen, anpassen	**WT-Strategien (Basisabsicherung)** - Anwenderfreundlich - Neue Arbeitsplätze im Bereich der Wartung

Da es bisher keine autonomen Industrieputzroboter gibt, ist die Marktzugänglichkeit gewährleistet und hohe Wachstumsraten sind möglich. Speziell bei Logistikbetrieben, Lagerhallen, Werkstätten und Produktionsbetrieben gibt es Potential bei der Optimierung der Reinigungskosten. Die Akzeptanz kann beim Management als Stärke gewertet werden. Die Modifizierbarkeit bietet sich zudem vor allem im Premiumsegment an. Aus den Stäken und Chancen lässt sich eine Investitions-Strategie ableiten. So wäre die Marktführerschaft im Bereich der autonomen Industrieputzroboter denkbar. Die Orientierung durch das Tiefenbild der Kinect-Kamera würde ein effizienteres Putzen durch das Vorausschauen (auch in

Dunkelheit) ermöglichen. Durch ein benutzerfreundliches HMI kann die Marktführerschaft unterstützt werden. Eine weitere Abmilderung der Schwäche durch Bedrohung der Arbeitsplätze kann mit neuen Arbeitsplätzen im Bereich der Wartung erfolgen.

Weniger Akzeptanz werden Putzfirmen und Reinigungsbetriebe aufbringen, denn durch diese autonomen Putzroboter würden Arbeitsplätze bedroht werden. Das Thema Akzeptanz wird auch im Buch: „Service-Roboter-Visionen“ aufgegriffen. "Ein wichtiger Aspekt dabei ist auch die Akzeptanz dieser Systeme in der Öffentlichkeit. Diese Akzeptanz wird mit der Sicherheit steigern." (Schraft, 2004, S. 31)

In der Entwicklung kann auch das Budget von CHF 600.- zu den Schwächen gezählt werden. Ausgleichende Strategien ergeben sich durch die Kombination der Schwächen und Chancen. Durch das hohe Marktpotential können Skaleneffekte ausgenutzt werden, welche im folgenden Kapitel erläutert werden.

Im Bereich der Risiken sind die neuen Wettbewerber und die Ressourcenkosten wichtig. Durch das hohe Wachstumspotential und der Modifizierbarkeit wäre eine vorausschauende Vorgehensweise nötig, um Umbrüche frühzeitig zu erkennen.

2.7. Wettbewerbsvorteile

Durch die Basisversion und die zahlreichen Upgrades kann ein grosser Markt anvisiert werden. Die Schweiz gilt allgemein als innovativ, so auch die Unternehmer. Somit ist es möglich, schnell viele Abnehmer zu finden.

Die Orientierung erfolgt mittels Bilderkennung. Das heisst, dass keine weiteren Installationen wie Linien auf dem Boden oder Laserschranken nötig sind. Die Tiefenbildkamera gewährleistet auch eine Orientierung bei Dunkelheit. Zudem ist dadurch bei jedem Gerät auch eine Kamera Installiert, was als zusätzliche Überwachungskamera eingesetzt werden könnte.

Der Industrieputzroboter wäre der erste vollständig autonome Reinigungsroboter auf dem Markt. Dadurch wird eine Vorreiterstellung eingenommen. Konkurrenten müssten sich zuerst die Technologie aneignen. Vor allem die Orientierung per Tiefenbildkamera ist eine grosse Erneuerung.

Ein weiterer Vorteil ist, dass für einen Käufer weniger Personalkosten entstehen. In einer Werkstatt müssten die Arbeitnehmer weniger Zeit für Reinigungen aufwenden. Die Reinigungskräfte werden trotzdem nicht sofort verschwinden. Vorerst würden die autonomen Reinigungsroboter eher in grossräumigen Hallen eingesetzt. Der Sammelbehälter muss regelmässig geleert werden und Ecken können nur bedingt gereinigt werden. Hierfür wäre ein Saugaufsatz denkbar. Der Kobold VR100 (2012) funktioniert z.B. mit einer Saugvorrichtung. Dies zeigt die Realisierbarkeit. Als Upgrade wäre ebenfalls ein automatisches Ausleeren des Sammelbehälters erhältlich, wie es bei Saugrobotern laut Saugroboter-24 (2012) üblich ist. Neue Arbeitsplätze im Bereich der Wartung und Fertigung würden entstehen.

2.8. Skalen- und Mengeneffekt

Durch Abnehmer wie Migros können viele Produkte an ein Unternehmen verkauft werden. Wird mehr verkauft, kann mehr produziert werden, was wiederum die Kosten pro Stück reduzieren würde. Dieser Einsparungseffekt bei den Stückkosten beträgt gemäss der Boston Consulting Group mit der „Verdopplung der kumulierten Produktionsmenge“ konstant 20-30%. Dies ist auch in nebenstehender Abbildung ersichtlich. Die Formel für die Erfahrungskurve lautet: $K_n=K_{1n}^{-b}$

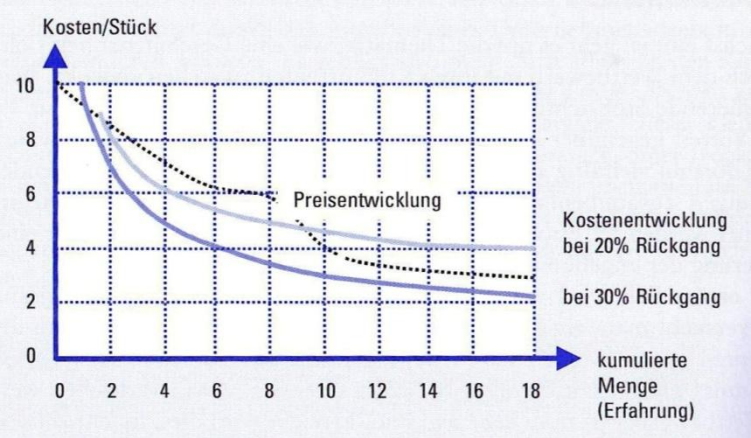


Abbildung 4: Erfahrungskurve

K_n: Stückkosten der n-ten Produktionseinheit
K_1:Stückkosten der 1. Protduktionseinheit
n: kumulierte Produktionsmenge
b: Degressionsfaktor,

(Müller-Stevens/Lechner (2011), S. 259/260)

Die Erfahurngseffekte können gemäss Prof. Dr. Hermann Jahnke der Universität Bielefeld dynamischer oder statischer Natur sein. Die dynamischen Faktoren sind va. auf Lerneffekte, technische Fortschritte und Rationalisierung zurückzuführen. Die statischen Faktoren betreffen die Fixkostendegression und den Betriebsgrösseneffekt. Somit ist ersichtlich, dass auch noch die Kosten stark senken werden, was den Roboter für den Markt viel erschwinglicher machen würde. Da die Bestimmung des Degressionsfaktors kompliziert und mit diversen Unsicherheiten verbunden ist, wird auf eine komplette Berechnung verzichtet. Die Grafik zeigt jedoch, dass die Kosten besonders am Anfang stark sinken werden, was so eine schnelle Etablierung im Markt als first-mover ermöglicht.

Entscheidet sich ein Abnehmer dafür mehrere Filialen mit solchen Robotern auszustatten, können Rabatte und/oder Dienstleistungen wie Wartung, Garantie und ähnliches angeboten werden. Ausserdem kann bei einem teuren Upgrade auch eine Wartungsgarantie mitgekauft werden. Der Anreiz eine grössere Investition zu tätigen wird somit erhöht.

3. Produktanforderungen

In PREN 1 wurde bereits eine Anforderungsliste erstellt. Auf diese aufbauend und ergänzend werden nun die Anforderungen an das marktfähige Produkt abgeleitet. Ein normaler Betrieb soll in allen Einsatzbereichen ermöglicht werden. Im Angebot sind zwei Basisversionen. Eine Kehrmaschine die leichter und günstiger ist. Diese wird vorwiegend in Lagerhallen, Werksbetrieben und ähnlichem zum Einsatz kommen. An diesen Orten ist es nicht nötig, dass der Boden feucht gewischt wird. Des Weiteren gibt es einen Scheuersauger, der grösser, schwerer und auch teurer wird. Das Einsatzgebiet des Scheuersaugers wird vorwiegend in Filialen von Detail- und Grosshändlern sein. Beide Versionen können mit verschiedensten Upgrades ergänzt werden.

3.1. Einsatzbereich/Umgebung

Der Einsatzbereich ist primär in Firmen und deren Lagerhallen bzw. Räumlichkeiten. Die Festforderungen (F) sind der „normale“ Betrieb unter „normalen“ Verhältnissen. Darunter wird verstanden, dass die Wetterbedingungen trocken sind und eine maximale Windstärke bis zu 4 Beaufort herrschen darf. Im „normalen“ Betrieb müssen alle Funktionen ohne Verzögerung ausgeführt werden können. Dazu gehören das Aufheben von Gegenständen, die Orientierung (auch bei Dunkelheit) und das autonome Ausführen von Aufgaben, wie Aufheben bzw. Einsammeln von Gegenständen und Schmutz. Der Betrieb in privaten Haushalten gehört zu den Wunschforderungen (W), da diese nicht primär zum gewählten A-Markt gehören. Die Mindestforderungen (M) sind der „normale“ Betrieb im Innenbereich und überdachten Aussenbereich.

Nr.	Art	Bezeichnung	Daten/Beschreibung	Verant-wortlich
1.	F	Betriebe	Aufgaben wie Aufheben von Objekten und Reinigen von Oberflächen/Böden	T
2.	W	Private Haushalte	z.B. mit den Aufgaben Aufräumen, Transportieren, Spielen/Unterhaltung	T
3.	M	Überdachter Aussenbereich	Bei „normalen“ Wetterverhältnissen dh. Windstärke bis zu 4 Beaufort	T
4.	W	Aussenbereich	Bei trockenem Wetter und Wind bis zu 4 Beaufort	T
5.	M	Innenbereich	Normaler Betrieb	T
6.	F	Boden	Linoleumboden, Parkett (nur Kehrmaschine), Asphalt	P,H,S
7.	M	Temperatur	0 – 40˚C	P,H
8.	M	Luftfeuchtigkeit	Nicht kondensierend, max. 80%	P
9.	W	Lichtverhältnisse	Max. 5000lx bei homogener Verteilung	P
10.	W	EMV	Störfestigkeit nach EN 61000-6-1:2007 Störaussendung nach EN 61000-6-3:2007	H

3.2. Beispielumgebungen und Programme

Der autonome Kehr- und Scheuersauger ist für verschiedene Einsatzgebiete programmierbar. Es gibt drei verschiedene Programme:

- Auto-Programm
- Auto-Programm mit Raumspeicherung
- Konfiguration von Räumen

Das Auto Programm ermöglicht die konfigurationsfreie Nutzung des Putzroboters. Der Putzroboter kann die Umgebung mit einem Algorithmus, welcher die Umgebung schrittweise erkennt, aufbauen. Im Vergleich zum Hauptkonkurrenten Kobold VR100, welcher den Raum per Laser-Erkennung aufbaut und eine Programmierung benötigt, ermöglicht der entwickelte Putzroboter eine sofortige Nutzung. Für fortgeschrittene Nutzer, die feste Flächen programmieren möchten, was in den ersten Reinigungszyklen einen Performance-Gewinn ermöglicht. Das Auto-Programm wird dann verwendet, wenn die Räume oft umgestellt werden und immer neue Hindernisse auftreten. Das Auto-Programm mit Raumspeicherung kommt zum Einsatz, bei wenig variablen Räumen wie z.B. Lagerhallen mit im Boden verankerten Regalen. Auch eine Konfiguration von Räumen ist in der mitgelieferten Konfigurationssoftware möglich, so kann schon beim ersten Einsatz die bestmögliche Route genutzt werden. Dies ist vor allem bei verwinkelten Räumen mit Gängen von Vorteil. Ein Objekterkennungsalgorithmus ermöglicht die Wahl einer effizienten Route wie in Abbildung 5. Das Projekt Mimicry (de.engadget.com) zeigt, dass mit der Kinect Räume in 3D eingescannt werden können. Die Technologie stammt aus dem Gameinterfacedesign. Diese neuartige Technologie lässt sich auch bestens für die Raumerkennung bei Putzrobotern nutzen. Nachfolgend drei typische Umgebungen für welche sich die Reinigungsprogramme unterschiedlich gut eignen.

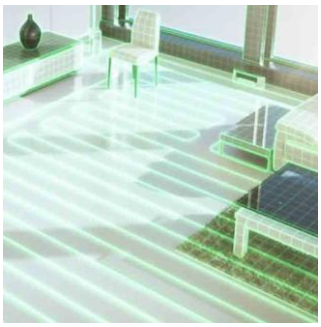


Abbildung 5: Raum-erkennung mit Route (aktion.kobold.vorwerk.com)

3.2.1. Lagerhallen

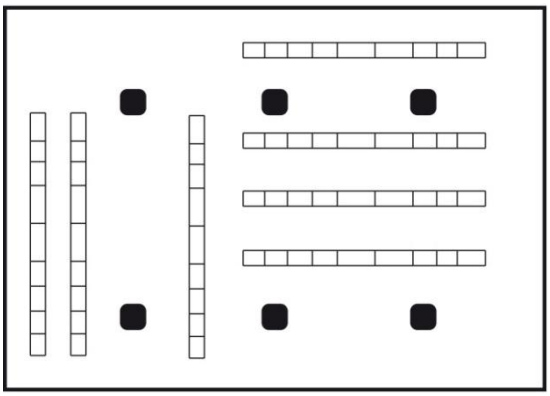


Abbildung 6: Grundriss Lagerhalle

Abbildung 7: Lagerhalle (123rf.com)

Im Bereich der Lagerhallen eignet sich vor allem das Auto-Programm mit der Raumspeicherung. Dies hauptsächlich, da die Bereiche sehr übersichtlich, wenig verwinkelt und die Wege lang sind. Auch wäre eine Konfiguration per Konfigurationssoftware denkbar, da die Regale fest mit dem Boden verschraubt sind. Die schwarzen „Flecken“ stellen Säulen dar, die sich ebenfalls nicht ändern werden.

3.2.2. Produktionshalle

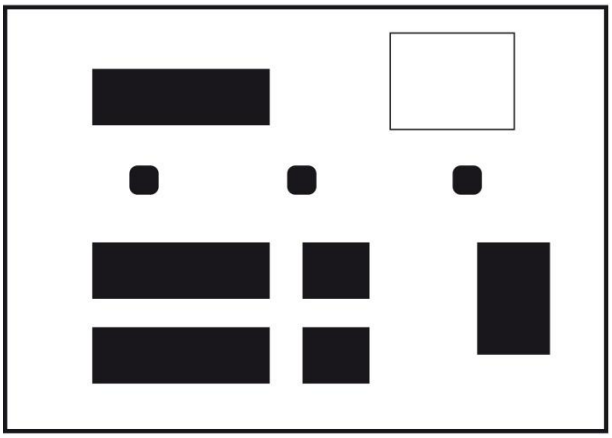


Abbildung 8: Grundriss Produktionshalle

Abbildung 9: Produktionshalle (theaustin.com)

In Produktionshallen ist ebenfalls die automatische Kartierung nützlich, wobei die Einstellungen natürlich auch mit der Konfigurationssoftware vorgenommen werden könnten. Das weisse Rechteck zeigt typische Abstellflächen. Bei diesen kontrolliert der Reinigungsroboter automatisch, ob sich Dinge auf der Fläche befinden, um diese gegebenenfalls zu umfahren. Bei der Produktionshalle auf Abbildung 9: Produktionshalle (theaustin.com) zeigt sich das Problem, dass die Höhe der Produktionsstrasse zu niedrig ist, um dort reinigen zu können. Auch hierfür gibt es eine Lösung: Ein optionaler Wischarm, der den Schmutz mit einer Wischbewegung hervor holt. Das Funktionsprinzip ähnelt dem eines Scheibenwischers. Für die glatte Oberfläche eignet sich der Scheuersauger besonders gut.

3.2.3. Werkstatt

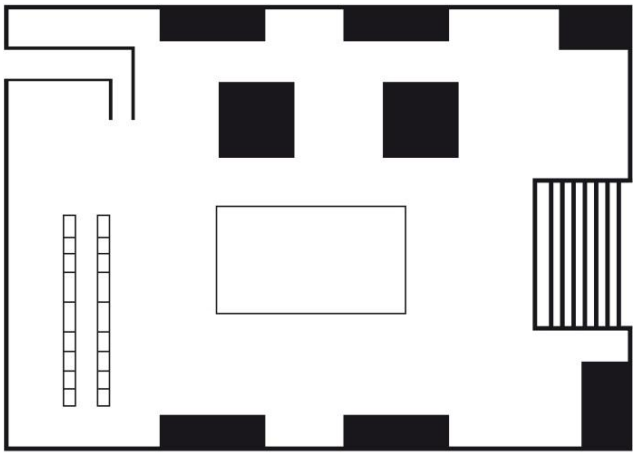


Abbildung 10: Grundriss Werkstatt

Abbildung 11: Werkstatt (metallbau-boehringer.de)

Diese Beispielwerkstatt weist typische Merkmale auf: festmontierte Regale, Lagerflächen in der Mitte, einer Treppe am Ende und einige feste Arbeitsplätze bzw. Werkbänke. Zudem wurde im Grundriss noch ein Gang eingebaut. Hier würde sich eine Konfiguration per Software lohnen, da so das Reinigen effizienter gestaltet werden könnte. Die Schwierigkeit besteht in der Unterscheidung zwischen den veränderlichen Hindernissen und den festen. Stufen und Treppen können mittels Bodensensor erkannt werden. Die Reinigung mit der Kehrmaschine wäre hier von Vorteil, da es sich beim Schmutz primär um Holzstaub handelt.

4. Abweichungen Prototyp / marktfähiges Produkt

Durch die Vorarbeit, die in PREN 1 und 2 gemacht wurden, steht die Grundfunktion der Lager- und Hallenreinigung. In diese Richtung fliesst also kaum mehr Entwicklungsaufwand. Trotzdem muss in einen marktfähigen Prototyp erheblich mehr investiert werden, als für den PREN-Roboter. Zusätzliche Baugruppen wie Auffangbehälter für Schmutz müssen noch eingebaut werden. Ausserdem braucht es noch Warnleuchten, Knöpfe und Schalter für ein sicheres Handling. Die Laufzeit müsste erheblich gesteigert werden und die Zuverlässigkeit muss gesichert sein. Auch die Konfigurationssoftware bedarf Anpassung, damit diese Marktfähig wird.

Nr.	Art	Bezeichnung	Daten/Beschreibung	Verant-wortlich
11.	F	Arbeitsstunden Werkstattpersonal	Max. 30h Elektrowerkstatt Max. 50h Maschinentechnik	H
12.	W	Arbeitsstunden Teammitglieder	300h PREN1 pro Teammitglied 300h PREN2 pro Teammitglied	T
13.	W	Gehäuse-Fertigung	40h	H
14.	F	Budget HSLU Budget für Prototypen	max. CHF 600.-, CHF 50’000.-	T

4.1. Produktion / Verkauf

Für das marktfähige Produkt müssen diverse Anpassungen vorgenommen werden. Das ganze Gefährt wird grösser und muss mit zusätzlichen Features ausgestattet werden. Durch höhere Stückzahlen kann der Arbeitsaufwand trotzdem stark verringert werden. Ein Putzroboter sollte weniger als 20 Arbeitsstunden für die Produktion beanspruchen. Die Basisversion soll in grösseren Mengen an verschiedene Abnehmer verkauft werden können. Der abgebildete Scheuersauger ist ab 3000.- € erhältlich. Der autonome Putzroboter soll bei derselben Grösse ab ca. 8'000.- CHF erhältlich sein. Weil der Roboter seine Arbeit autonom macht, ist die Investition nach kurzer Zeit wieder kompensiert. Neben dem Produkt selber werden ausserdem Dienstleistungen wie Wartung, Einführung, und Garantieleistungen angeboten.

Abbildung 12: Scheuersaugmaschine (reinigungsberater.de)

Nr.	Art	Bezeichnung	Daten/Beschreibung	Verant-wortlich
15.	W	Verkaufspreis	Ab CHF 7'993.-	W
16.	M	Qualität	Max 2% fehlerhafte Produkte	T
17.	M	Herstellerkosten	CHF 5000.- bei einer Serie von 500 Stück	T

4.2. Fahrzeugeigenschaften und Funktionen

In nachfolgender Tabelle werden die wichtigsten Eigenschaften und Masse des autonomen Industrieputzroboters aufgeführt und erläutert. Grossen Wert wird auch auf die Sicherheit gemäss den Europäischen Sicherheitsnormen für Maschinen (2007) gelegt.

Nr.	Art	Bezeichnung	Daten/Beschreibung	Verant-wortlich
18.	W	Akkulaufzeit	4 Stunden	H
19.	M	Akkulaufzeit	2 Stunden	H
20.	M	Aufladen	Automatisches Aufsuchen der Ladestation bei Stromknappheit	H
21.	F	Anpassbarkeit	Möglichkeit zur nachträglichen Montage eines Saugaufsatzes, 360° Kameras, Bürsten, Zusatztanks (für Schmutz), grössere Akkus, Sensoren usw. usf.	T
22.	F	Abmessungen für Kehrmaschine	max. 100cm breit max. 100cm lang max. 70cm hoch	H
23.	F	Abmessungen für Scheuersauger	max. 90cm breit max. 140cm lang max. 100cm hoch	H
24.	F	Verlässlichkeit	Akkulaufzeit eingehalten, stabiles Gehäuse, wenn eine Person bis zu 100kg dagegen läuft, muss der Betrieb wieder ordnungsgemäss aufgenommen werden können.	H
25.	F	Betriebssicherheit	Einhalten von Normen gemäss den Europäische Sicherheitsnormen für Maschinen (2007), Not-Stop bei Fehler usw.	H
26.	M	Gewicht der Kehrmaschine	max.30 kg (ohne Erweiterungen)	H
27.	M	Gewicht der Scheuermaschine	max.100 kg Leergewicht	H
28.	M	Lautstärke	Max. 67dB bei 20Hz – 20kHz	H
29.	F	Reinigungsarbeiten	Durchführen von Reinigungsarbeiten am Boden (trocken oder nass)	H
30.	F	Gegenstände aufheben	Bis zu einer Grösse von (L*B*H) 7cm*7cm*30cm (Je nach Einsatzgebiet und Aufsatz)	H
31.	F	Autonomie	Keine physische Verbindung zum Fahrzeug (z.B. für die Speisung / Steuerung)	T
32.	F	Orientierung bei Dunkelheit	Keine physische Verbindung zum Fahrzeug (z.B. für die Speisung / Steuerung)	T
33.	F	Objekterkennung	Objekte müssen erkannt werden.	S

4.3. Bedienung / Einstellbarkeit

Die Bedienung soll am Gerät selber und extern an einem Computer über W-LAN möglich sein. Die aufgezeichneten Daten können direkt auf dem Single-Board-Computer oder auf dem externen Steuerungscomputer gespeichert werden. Fehler während des Betriebs können am PC und ortsunabhängig behoben werden. Die Konfigurationssoftware ist für verschieden Szenarien und je nach Gebrauch vorprogrammiert. Somit kann sich der Roboter in praktisch allen Hallen völlig autonom zurechtfinden.

Nr.	Art	Bezeichnung	Daten/Beschreibung	Verant-wortlich
34.	F	Stoppen des Fahrzeugs	Autonomes Stoppen	S
35.	F	Starten des Fahrzeugs	Einfache Bedienung für Startsequenz am Fahrzeug oder am PC	H,S
36.	M	Einstellbarkeit	über eine Konfigurations-Datei am PC	S
37.	M	Einstellbarkeit	über ein Programm mit visueller Darstellung des Parcours	S
38.	M	Steuerungs-Software	Grundsteuerung anhand der Konfigurations-Datei für jede Umgebung anwendbar. Individuell auf Kundenwunsch programmierbar.	S

4.4. Erweiterbarkeit

Die Basisversion ist als Trockenkehrmaschine oder als Scheuersaugmaschine erhältlich. Beide Versionen können individuell und beinahe unbegrenzt erweitert werden.

Nr.	Art	Bezeichnung	Daten/Beschreibung	Verant-wortlich
39.	W	Kamera	durch zusätzliche Kamera kann ein 360° Blickfeld erreicht werden	H,S
40.	W	Aktoren	durch zusätzliche Aktoren erweiterbar	H,S
41.	W	Softwarefunktionen	durch zusätzliche Updates erweiterbar	S
42.	W	Steuerungs-Software	je nach Einsatzbereich speziell programmierbar	S
43.	W	Zusätzliche Akkus	längere Laufzeit, mehr Leistung	H
44.	W	Zusätzlich Auffangbehälter	muss weniger geleert werden, langer Einsatz	H
45.	W	Sensoren	neben der Tiefenkamera können zusätzliche Sensoren wie Infrarot, Laser oder Ultra-Schall angebracht werden.	H

5. Designstudie

Das Design hat eine unterstützende Funktion und soll einen Mehrwert für das Produkt schaffen. Hierbei spielt die Ästhetik eine wichtige Rolle. Wobei zwischen Objekt- und Subjektästhetik zu unterscheiden ist. Weitere Gestaltungsprinzipien sind aufgabengerechte Dimensionierung, Einfachheit, Eindeutigkeit, Führungshilfen und Konsistenz der Gestaltung. Ausserdem sollte das Produkt möglichst selbsterklärend sein. Auf Grund der verwendeten Kinect und der Konsistenz wurde die Formgebung von der X-Box inspiriert. Deshalb wird das Ganze technomorph gestaltet. Ausserdem sind bisher erhältliche Kehrmaschinen und Scheuersauger eher technomorph gestaltet.

Abbildung 13: x-box 360 mit Kinect[1]

5.1. Anforderungen an das Design

Das Gefährt soll leicht und trotzdem stabil sein. Aus diesem Grund wurde ein Gehäuse aus schwarz eingefärbter Glasfaser gewählt. Ausserdem soll ein leichter Zugang an die gesamte Elektronik gewährleistet sein. Durch eine abnehmbare Haube und weiteren Zugängen am Unterboden ist eine einfache Wartung möglich.

5.2. Designanalyse

Abbildung 14: Neato und Roomba von iRobot (www.blogcdn.com)

Das Design von Robotern ist abhängig von deren Einsatzgebiet, Funktionen und Komponenten. Auf nebenstehender Abbildung sind die zwei meistverkauften Reinigungsroboter abgebildet. Links der Neato, welcher über einen Laser verfügt und so Räume einscannen kann. Der Roomba (rechts) verfügt über Drucksensoren und reinigt nach dem Zufallsprinzip. Beide Roboter zeigen eine gute Übersicht über das

[1] http://articles.businessinsider.com/2010-12-10/tech/30000897_1_kinect-xbox-controllers

übliche Design. Die meisten Roboter sind rund und ähnlich wie der Roomba aufgebaut. Roomba verfügt wie die meisten anderen über ein futuristisches Design. Neatos Design ist eher zweckmässig und erzeugt nur wenig Emotionen. Jedoch ist das Design bei beiden konsistent. Die bedienbaren Knöpfe lassen sich jeweils schnell erkennen.

5.3. Bestehende Modelle von Scheuersaug- und Kehrmaschinen

In diesem Unterkapitel werden die Designs von einigen wichtigen Herstellern im Bereich der Gebäudereinigung gezeigt.

Abbildung 15: Wetrok Kehrmaschine (wetrok.ch)

Wetrok setzt beim Design immer dieselben Farben ein. Das sorgt für Konsistenz in der Produktreihe und ihre Maschinen und Geräte sind dadurch auch schon von weitem erkennbar und als Produkte von Wetrok auszumachen.

Der Scheuersauger von Kärcher ist im Gegensatz zu Wetrok gelb. Auch hier sorgt die Farbe dafür, dass die Maschine von weitem als Kärcher-Produkt ausgemacht werden kann. Die zweite Farbe ist allerdings wie bei Wetrok anthrazit. Der Rest des Designs unterscheidet sich nicht gross. Würde man die Farbe des einen Geräts an das andere anpassen, könnte man erst bei näherem Betrachten - oder sogar erst beim Lesen der Aufschrift erkennen, welches Produkt von welchem Hersteller stammt.

Abbildung 16: Kärcher Scheuersaugmaschine (kärcher.ch)

5.4. Entwürfe für PREN

Da als Sensor eine Kinect-Kamera von Microsoft favorisiert wurde, mussten die Entwürfe von PREN1 angepasst werden, um den Anforderungen der Kinect zu genügen. Die Formsprache wurde aufgegriffen und in einem technomorphen Aussehen skizziert, um die Kinect-Kamera optisch gut zu integrieren. Durch die Tatsache, dass Kinect erst brauchbare Bilder ab einem Abstand von ca. 50 cm liefert, ist es vorteilhaft, diese im hinteren Bereich des Fahrzeugs anzubringen. Ebenfalls wurde anschliessend Platz für die einzelnen Komponenten des autonomen Schnelltransporters eingeplant. Nachfolgend wird der Entwurfsprozess dargestellt.

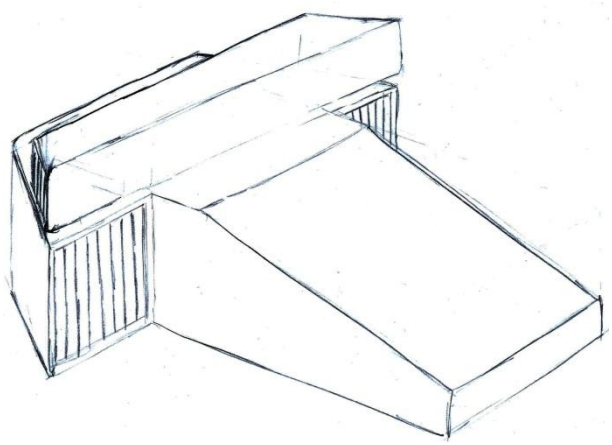


Abbildung 17: Entwurf Gehäuse für PREN

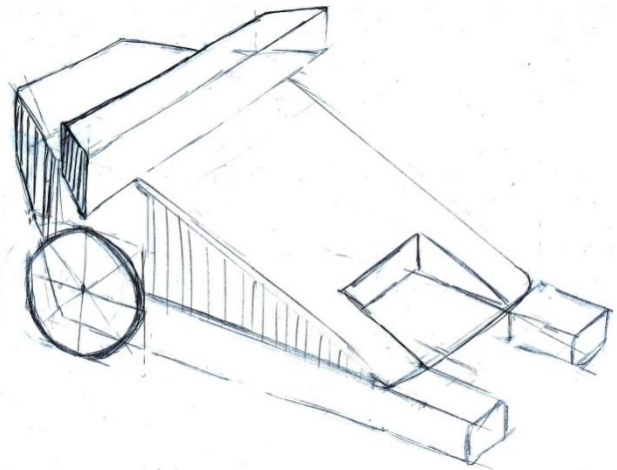


Abbildung 18: Entwurf Fahrzeug für PREN

Das fertige Produkt für PREN sieht wie in Kapitel 5.1 erklärt folgendermassen aus:

Abbildung 19: Fahrzeug von der Seite und von vorne

5.5. Entwürfe für Markt

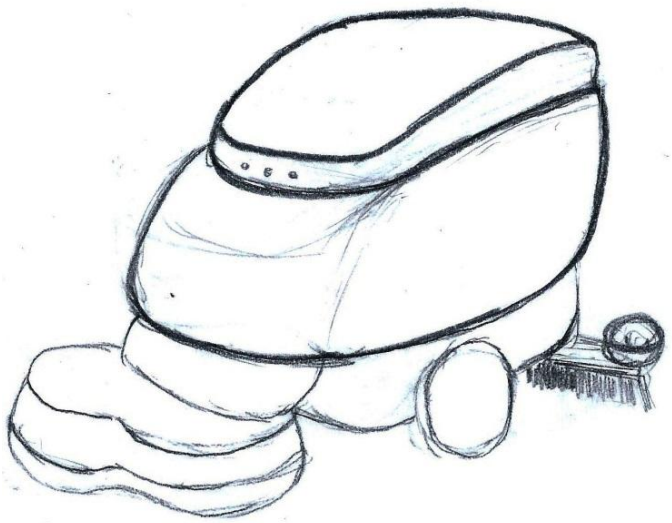


Abbildung 20: Entwurf für Marktanwendung gross

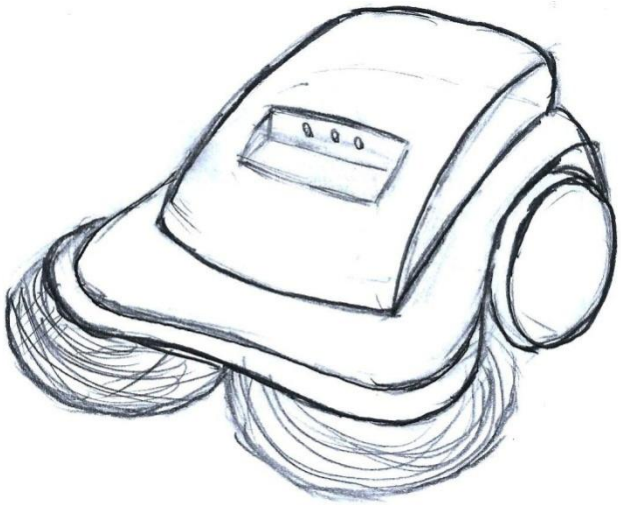
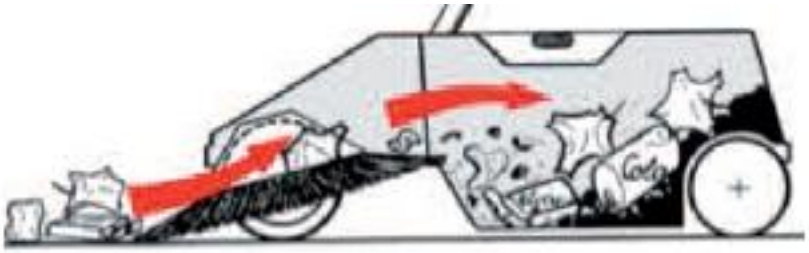


Abbildung 21: Entwurf für Marktanwendung klein

Bei den grösseren Scheuersaugern ist schon genügend Platz für die zusätzliche Elektronik vorhanden. Da die Bilderkennung erst ab einem gewissen Abstand funktioniert, werden die Kameras etwas höher und zurückversetzt eingebaut. Ein Vorteil kann durch schwenkbare Scheuerbürsten geschaffen werden. Damit kommt man besser in die Ecken. Das kleinere Modell ist die Kehrmaschine. Diese haben bisher nur wenig Platz für zusätzlich Komponenten wie Motoren und Hardware. Ein zusätzlicher Aufbau wäre hier eine Lösung. Auch hier müssten Kameras weiter hinten montiert werden. Das Prinzip bleibt das gleiche.



Mit den Tellerbürsten wird der Schmutz in den Sammelbehälter im hinteren Bereich befördert.

Abbildung 22: Kehrprinzip (www.wetrok.ch)

5.6. Beispiele vom Markt

Abbildung 23: Kehrmaschine von Haaga (industriebedarf-staplerteam.de)

Die Kehrmaschine von Haaga weist zweckorientiertes Design auf. Durch die fast gleiche Höhe des gesamten Gefährts wird viel Platz bewahrt, der bei anderen Designs verschenkt wurde. Auf Abb. 19 ist zu sehen, dass es im vorderen Bereich aber gar keinen zusätzlichen Platz braucht.

Der Scheuersauger von Tennant hat eine runde Form. Die Farbgebung erinnert an Wetrok, ist jedoch matt und das Grau ist heller. Durch die grössere Anzahl an Komponenten als bei Kehrmaschinen, sind Scheuersauger in der Regel grösser. Viele Modelle sind ausserdem mit Batterien ausgestattet und brauchen keinen Stromanschluss während des Betriebs.

Abbildung 24: Scheuersaugmaschine von Tennant (directindustry.de)

Abbildung 25: Concept Car (nzz.ch)

Neue Designstudien von Fahrzeugherstellern gehen wieder auf abgerundete Formen zu. Es zeigt sich also, dass solche Modelle in Zukunft häufiger werden. Obwohl eckige Formen mehr Platz gewähren, wird mit runden Modellen gearbeitet. Folglich kann damit gerechnet werden, dass sich diese runden Formen auch bei anderen Geräten durchsetzen.

Nachdem die Designanalyse für den Markt abgeschlossen war, stellte sich heraus, dass im Bereich der Reinigungsmaschinen nicht viel Abwechslung besteht. Ändert man die Farben, können die Marken kaum noch zugeordnet werden. Es gibt jedoch Formtechnisch einige Punkte, die noch verbessert werden können. Die Kehr- und Scheuerbürsten können besser in das Gehäuse integriert werden. Die meisten autonomen Saugroboter weisen eine runde Form auf. Diese suggeriert zwar Wendigkeit, jedoch wird so Platz verschenkt, welcher für den Auffangbehälter und genügend Leistung gebraucht werden kann. Speziell im B2B-Bereich, wo die Räume grösser sind, fällt dies ins Gewicht. Herkömmliche Modelle besitzen oftmals eine knappe Laufzeit. Der Roboter für den Markt wird speziell für den B2B-Markt optimiert.

6. Beschreibung des Prototyps

In diesem Kapitel soll eine Übersicht über den hergestellten Prototyp und die Bewältigung des PREN-Parcours gegeben werden.

6.1. Übersicht

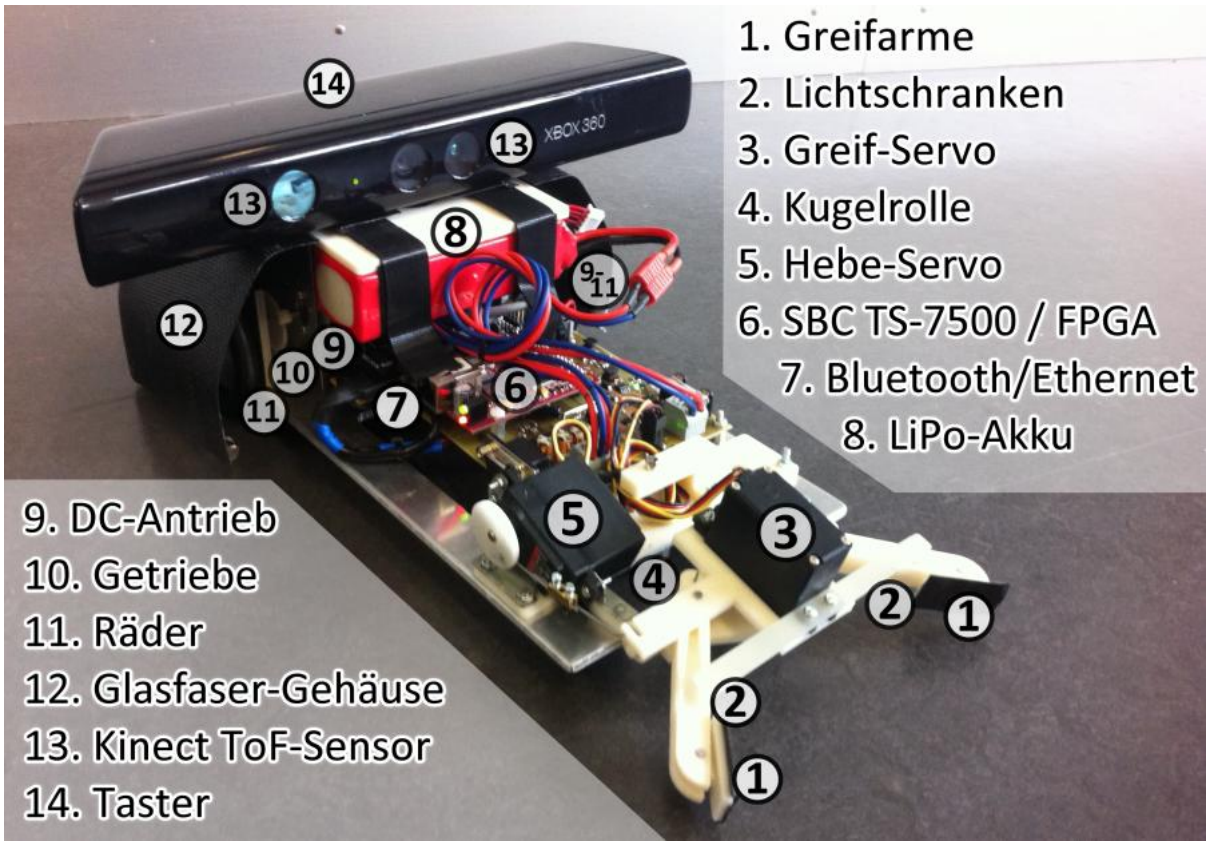


Abbildung 26: Übersicht Funktionsmuster

6.1.1. Hardware

Der Roboter wird als zweirädriges Fahrzeug mit einer Kugelrolle als Vorderstütze realisiert. Er wird über eine Panzersteuerung an den Hinterrädern gelenkt. Dazu dienen zwei DC-Motoren. Die Würfelergreifung erfolgt mittels einer Zange mit zwei Greifarmen, die durch einen Servo-Motor betrieben werden. Das Gehäuse besteht aus schwarz eingefärbter Glasfaser. Der vordere Teil ist mit Magneten befestigt und kann abgenommen werden.

Zur Objekt- und Positionserkennung wird die Tiefenkamera der Microsoft Kinect Sensorleiste eingesetzt. Diese liefert ein Tiefenbild (640x480x11Bit) des Sichtfeldes ab 0.45m bis 6m. Die Leiste wird am hinteren Fahrzeugteil möglichst tief montiert. Zur Wegmessung werden zwei Encoder an den Rädern eingesetzt. Als Stromversorgung dient ein Lithium-Polymer-Akkumulator mit 4 Zellen und einer Nennkapazität von 2200mAh.

6.1.2. Software

Die zentrale Recheneinheit des Fahrzeugs ist ein Single Board Computer (TS-7500 von embeddedARM, 250Mhz ARM9 CPU, 64MB DDR-RAM, 4MB Flash, 5K FPGA, 2xUSB, Ethernet). Als Betriebssystem ist bereits ein Debian Linux vorinstalliert. Die Steuerungssoftware läuft eigenständig auf dem Single Board Computer (SBC). Sie beinhaltet getrennte Software-Module zur Ablaufsteuerung, Bilderkennung, Navigation und Antriebssteuerung. Zur Bewegungsregelung wird der vordefinierte Pfad mit einem virtuellen Liniensensor geschnitten. Die Regelgrösse für den PID-Regler ist die Richtungsabweichung.

Damit die Software auch ohne Hardware getestet werden kann, wurde ein Emulator geschrieben, der die Physik und das Kinect-Tiefenbild berechnet. Zur Trennung von Programmlogik und Hardware-Ansteuerung wird eine Treiberschicht (Hardware abstraction layer, HAL) implementiert. Die Programmlogik kann dann entweder mit den Treibern zur Ansteuerung der realen Komponenten oder mit dem Emulator zusammengeschaltet werden. Für die Kinect-Kamera wird der Open-Source-Treiber libfreenect verwendet.

6.1.3. Konzept-Änderungen

Zur Konfiguration des Roboters war ursprünglich die Erstellung eines Programms zur grafischen Eingabe des Parcours geplant. Darauf wurde aus Zeitgründen verzichtet. Die Einstellungen zur Objekterkennung sowie allgemeine Optionen werden stattdessen in einer Konfigurationsdatei gespeichert. Das Verhalten und der zu fahrende Weg sind hartcodiert, jedoch auf einem relativ hohen Abstraktionsniveau implementiert.

Neu in PREN2 wurde eine Lichtschranke am Greifer montiert, die erkennen soll, wann der Würfel aufgehoben werden kann. Ausserdem dient ein Taster am Gehäuse zum Starten des Programms.

6.2. Parcours-Ablauf

Der PREN-Parcours wird wie folgt abgearbeitet (Abbildung 27):

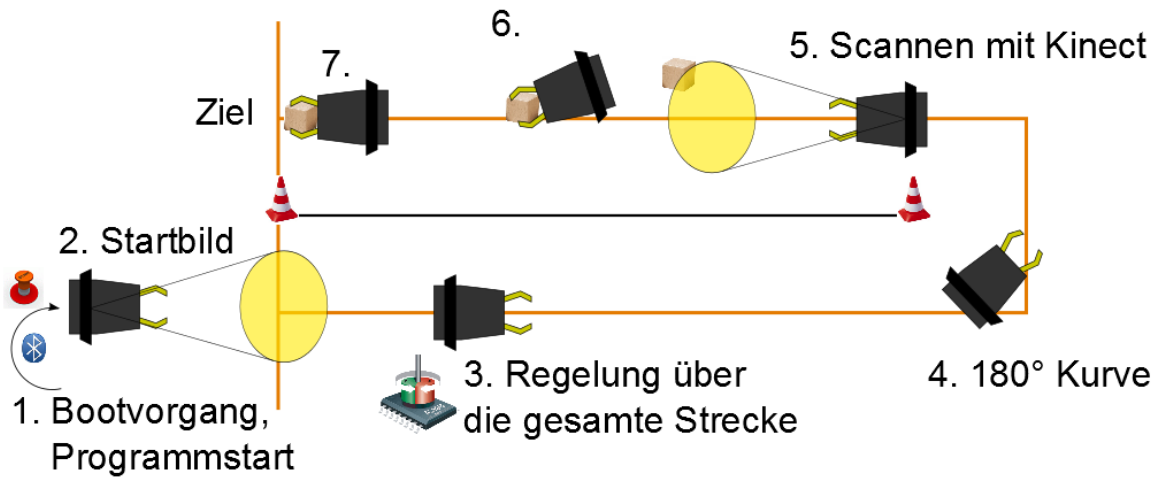


Abbildung 27: Ablauf des Parcours

1. **Betriebssystem und Programm starten.**
   Das Linux-System bootet und startet die Bluetooth-Verbindung. Das Programm kann entweder über Taster oder Terminal (serielle Schnittstelle über Bluetooth) gestartet werden.
2. **Startbild aufnehmen und auswerten.**
   Ein Kinect-Tiefenbild wird aufgenommen und nach Pylon und Würfel untersucht. Daraus kann der Startwinkel korrigiert werden. Mit drücken des Tasters startet das Fahrzeug.
3. **Geradeaus fahren bis zum Kurvenpylon.**
   Das Fahrzeug fährt 4m geradeaus bis Höhe Pylon. Kurz davor wird für die Kurve verlangsamt.
4. **180°-Kurve fahren.**
   Der Pylon wird im Radius eines halben Meters umfahren.
5. **Würfel und Endpylon im Tiefenbild suchen.**
   Das Fahrzeug hält kurz nach dem Pylon an und nimmt ein Tiefenbild auf, um Endpylon und Würfel zu suchen. Die Suche beschränkt sich auf eine definierte Box im 3D-Raum. Sind mehrere Kandidaten vorhanden, werden diese nach Grösse, Form und Position bewertet.
6. **S-Kurve bis zum Würfel fahren und diesen aufheben.**
   Die S-Kurve wird so berechnet, dass das Fahrzeug kurz vor dem Ergreifen (100mm) nur noch geradeaus fahren muss. Erkennt die Lichtschranke am Greifer den Würfel, so wird er gepackt und gehoben. Ansonsten wird er zuerst noch max. 300mm geschoben.
7. **Ins Ziel fahren.**
   Je nach Position kann direkt geradeaus beschleunigt und ins Ziel gefahren werden. Ist das Fahrzeug zu nah am Rand oder der Mittellinie wird auf die Mitte des Ziels zugesteuert.

7. Software

7.1. Komponenten

7.1.1. Übersicht

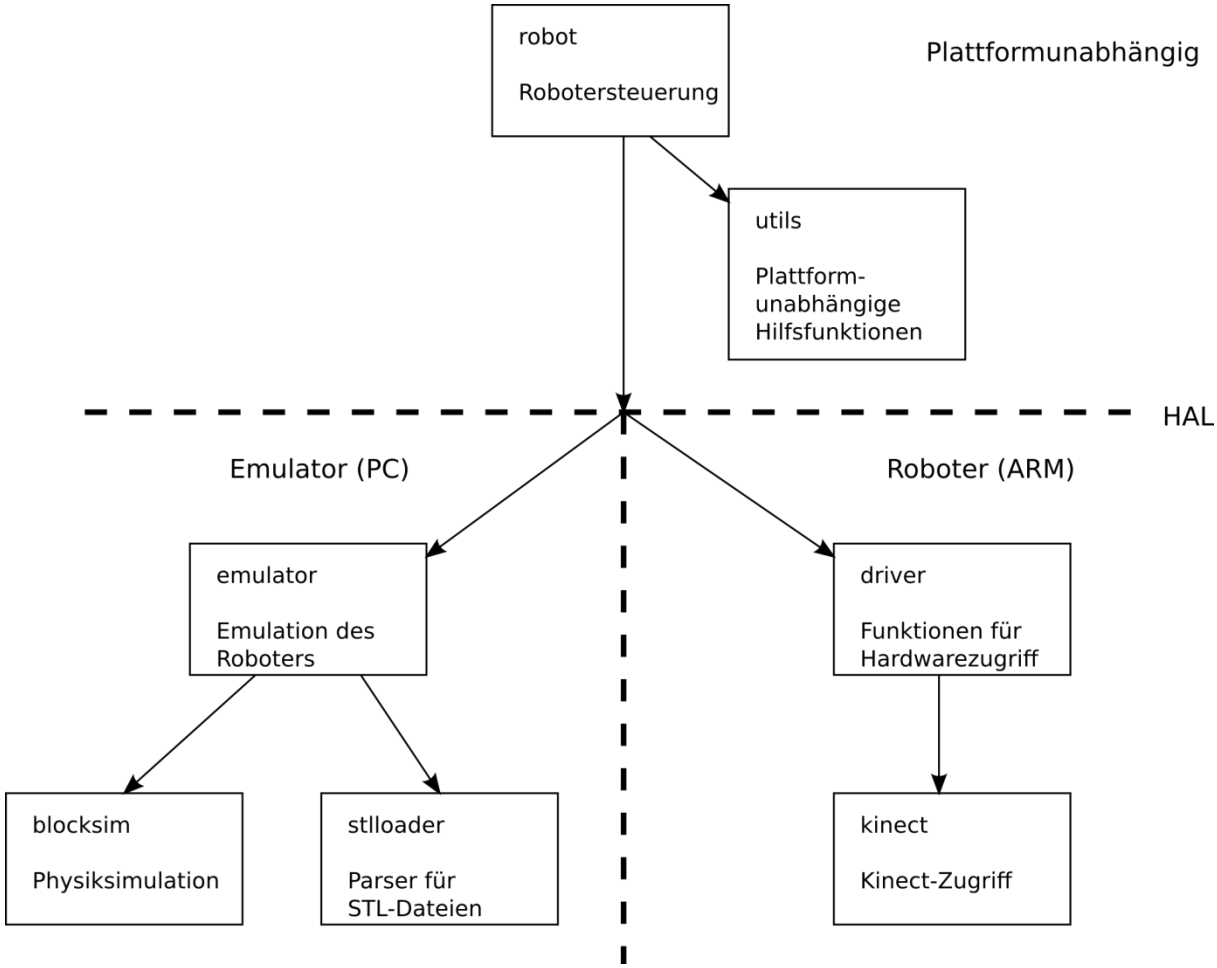


Abbildung 28: Grobe Sicht auf die Softwarearchitektur

Die Software kann in drei Hauptteile unterteilt werden:

- Einen hardwareunabhängigen Teil (oben), bestehend aus dem Hauptprogramm "robot", sowie der Bibliothek mit Hilfsfunktionen (utils)
- Einen Teil zur Ausführung auf einem normalen x86-Rechner in emulierter Form
- Einen Teil zur Ansteuerung der Roboterhardware zur Ausführung auf dem Singleboardcomputer mit der Zielarchitektur ARM

Sämtlicher Code ist in C/C++ verfasst und für die Kompilierung unter Linux mit der GCC (GNU compiler collection) vorgesehen. Für die Erstellung des Programms für den

Singleboardcomputer kann die Cross-Toolchain[2] von Technologic Systems verwendet werden. Weiterhin ist es möglich, den Code direkt auf dem SBC zu kompilieren.

Als Buildsystem wird GNU Make verwendet. Die Buildrules sind projektübergreifend erstellt, so dass Komponenten auch dann neu kompiliert werden, wenn sich ein Teil eines untergeordneten Projekts verändert hat. Aus diesem Grund kann eine beliebige IDE oder auch ein einfacher Texteditor zum Bearbeiten des Quellcodes verwendet werden.

Der Code des Emulators ist weitgehend betriebssystemunabhängig geschrieben, weshalb auch eine Ausführung auf einem anderen System als Linux (beispielsweise Windows) denkbar wäre. Dies wurde jedoch nicht weiter verfolgt.

7.1.2. Klassenübersicht

Das nachfolgende Klassendiagramm zeigt die Abhängigkeiten der einzelnen Softwaremodule.

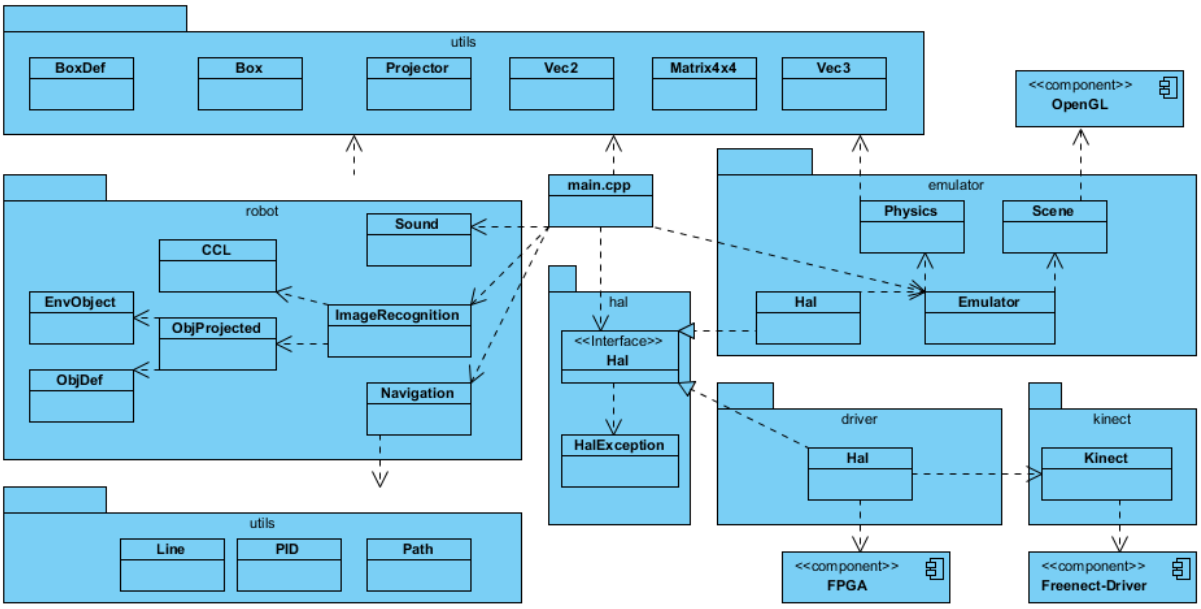


Abbildung 29: Klassendiagramm der Steuerungssoftware

[2] ftp://ftp.embeddedarm.com/ts-arm-sbc/ts-7500-linux/cross-toolchains/crosstool-linux-gcc-4.5.2-gclibc-2.9-oabi.tar.gz

7.1.3. robot

Das Teilprojekt "robot" enthält die eigentliche Robotersteuerung. Hier ist die Programmlogik zum Absolvieren des Parcours enthalten, sowie die dazu benötigte Bewegungssteuerung und die Bilderkennung. Bei der Kompilierung dieses Teilprojekts entsteht die ausführbare Datei "robot", welche die gesamte Intelligenz des Roboters enthält. Alle Unterprojekte sind als statische Bibliotheken dazugelinkt.

7.1.4. utils

Diese Komponente enthält einige Hilfsklassen mit allgemein einsetzbaren Algorithmen und Datenstrukturen:

- Zwei- und dreidimensionale Vektoren mit den gebräuchlichen Operationen
- 4x4-Matrix für Koordinatentransformationen
- Projektor-Klasse zur Projektion von Koordinaten von 2D nach 3D und umgekehrt
- Allgemeine Linien-Klasse
- Pfad-Klasse zur Darstellung eines Pfades aus einzelnen Linienstücken
- PID-Regler

7.1.5. hal

Der HAL (Hardware Abstraction Layer) dient als Schnittstelle zwischen dem plattformunabhängigen Code und den darunterliegenden Komponenten. Die Klasse Hal wird einmal vom "emulator" und einmal vom "driver" implementiert.

7.1.6. emulator, blocksim, stlloader

Diese Teilprojekte stellen den Hardwareemulator dar. "stlloader" enthält einen Parser für STL-Dateien, womit 3D-Modelle aus dem CAD-System eingelesen und mittels OpenGL[3] gerendert werden können. "blocksim" ist ein Simulationsframework womit Blockdiagramme mittels Funktionen und Signalen beschrieben und anschliessend simuliert werden können. Damit wird das physikalische Verhalten des Roboters modelliert.

Der "emulator" ist schliesslich für die Darstellung der ganzen 3D-Szene verantwortlich und stellt die PC-basierte Implementierung des HAL zur Verfügung.

7.1.7. driver, kinect

Der "driver" ist das Gegenstück zum Emulator für den Roboter. Er implementiert den HAL mit den Funktionen zur Ansteuerung der Motoren, Servos, Encoder, etc. Das Teilprojekt "kinect" enthält den Treiber von OpenKinect[4] zur Verwaltung der Kinect-Kamera.

[3] http://www.opengl.org/
[4] http://openkinect.org/wiki/Main_Page

7.2. Bewegungssteuerung

7.2.1. Übersicht

Die Bewegungssteuerung/-regelung wird als Linienfolger ausgeführt. Weil jedoch kein "echter" Liniensensor vorhanden ist, wird eine virtuelle Linie konstruiert. Dies passiert beim Programmstart und ist gegeben durch den Aufbau des Parcours.

Abbildung 30 zeigt die Architektur der Bewegungssteuerung. Der Sollwert für den Richtungsregler ist immer 0, dies entspricht einer Ausrichtung des Fahrzeugs auf die Solllinie.

Abbildung 31 zeigt die konstruierte Linie im Emulator für den PREN-Parcours. Im ersten Schritt fährt der Roboter bis zum Ende der Linie und nimmt dort ein Bild auf. Nach der Auswertung ist die Position des Würfels bekannt und die Linie wird entsprechend erweitert.

Anhand der Encodersignale ist die aktuelle Position und der Winkel des Roboters bekannt, wodurch ebenfalls ein virtueller Liniensensor konstruiert werden kann. Dieser liefert anschliessend den IST-Wert für den Regler.

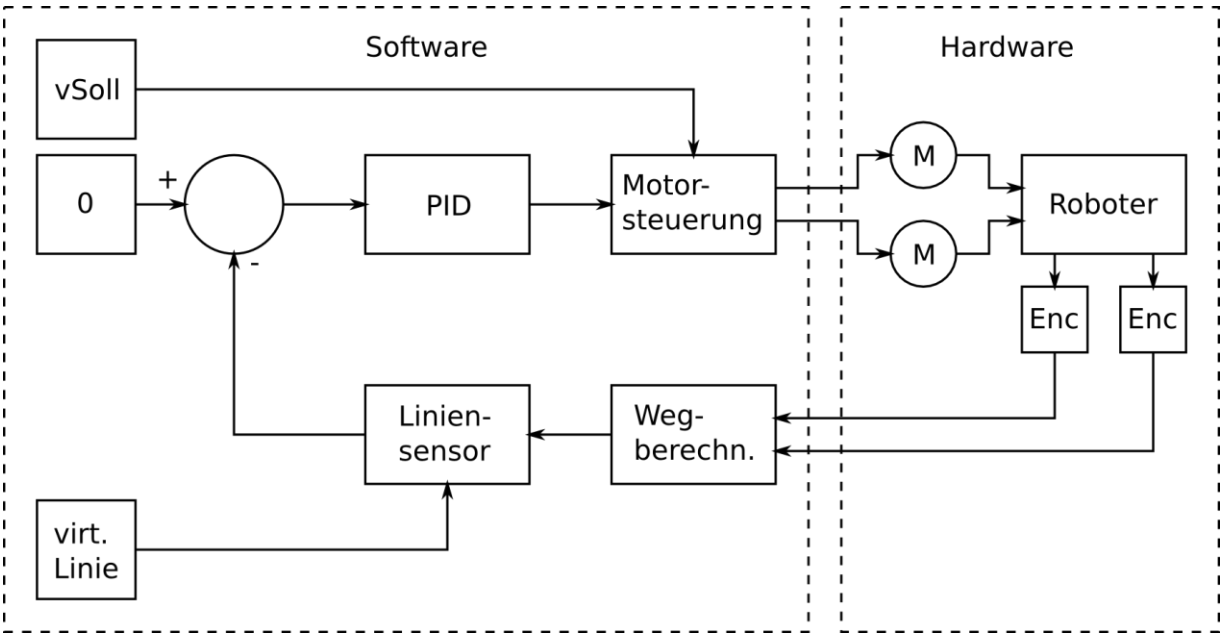


Abbildung 30: Blockschema des Regelkreises

7.2.2. Streckenführung

Die Vorgabe der Strecke wird zur einfacheren Verarbeitung als Liste von Vektoren mit geraden Verbindungsstücken ausgeführt. Wird der Liniensensor ebenfalls als Gerade interpretiert, so kann über die Schnittpunktberechnung relativ einfach auf den aktuellen IST-Wert für den Regelkreis geschlossen werden. Durch die Berechnung der Kreuzprodukte von aufeinanderfolgenden Richtungsvektoren lässt sich etwas über die Krümmung der Strecke aussagen, womit ein Sollwert für die Geschwindigkeit vorhanden ist.

Um den Rechenaufwand zu minimieren werden zu jedem Zeitpunkt nur die nächsten paar Streckenabschnitte ab der aktuellen Position ausgewertet. Die analysierte Länge sollte damit ca. 30cm betragen.

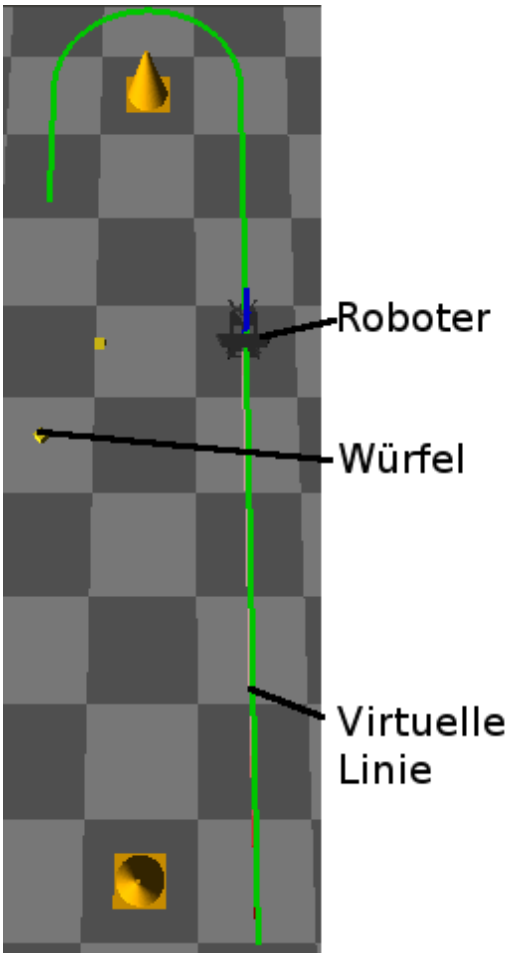


Abbildung 31: Virtuelle Linie im Emulator

7.2.3. Richtungsbestimmung

$$\vec{r_i} = \overrightarrow{P_{i+1}} - \vec{P_i}$$

$$\vec{r_n} = \frac{\vec{r_i}}{|\vec{r_i}|}$$

$$L_i(t) = \vec{P_i} + t \cdot \vec{r_n} \quad \text{mit} \quad t = [0, |\vec{r_i}|]$$

$$S(u) = \overrightarrow{S_0} + u \cdot \overrightarrow{S_r}$$

$$S(u) = L_i(t)$$

$$\Rightarrow u = \frac{P_x \cdot r_y - P_y \cdot r_x + r_x \cdot S_{0y} - r_y \cdot S_{0x}}{r_y \cdot S_{rx} - r_x \cdot S_{ry}}$$

$$\Rightarrow t = \frac{P_x \cdot S_{ry} - P_y \cdot S_{rx} - S_{0x} \cdot S_{ry} + S_{0y} \cdot S_{rx}}{r_y \cdot S_{rx} - r_x \cdot S_{ry}}$$

Bedingung: $0 \leq t \leq |\vec{r_i}|$

$\vec{P_i}$: Position des Wegpunkts i

$\overrightarrow{S_0}$: Mittelpunkt des Liniensensors in Absolutkoordinaten

$\overrightarrow{S_r}$: Ausrichtung des Liniensensors (90° zur Fahrtrichtung)

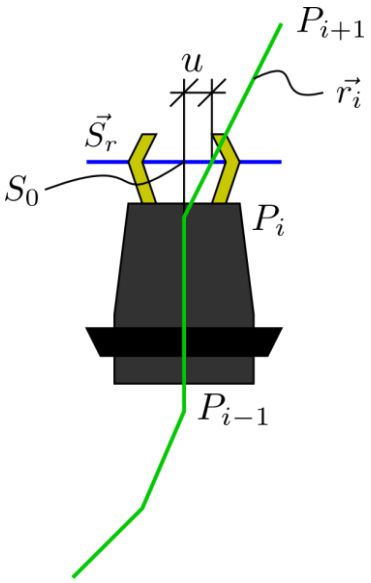


Abbildung 32: Richtungsbestimmung

Die Variable *u* enthält damit den Abstand des Schnittpunktes zum Mittelpunkt des Liniensensors und kann direkt als IST-Wert für den Positionsregler verwendet werden.

WICHTIG: Diese Regelung ist nur für Vorwärtsfahrt möglich. Bei Rückwärtsfahrt muss der virtuelle Liniensensor hinter das Fahrzeug platziert werden.

7.2.4. Geschwindigkeitsvorgabe

$$c = \sum_{k=i}^{i+n} |r_{k-1} \times r_k| \quad \text{bei} \quad \sum_{k=i}^{i+n} |r_k| \approx 30\text{cm}$$

$$v_{SP} = v_{max} \cdot K_v \cdot \frac{1}{c} \quad \text{mit} \quad c > 0$$

v_{SP} : Vorgabewert für Geschwindigkeitsregler/-steuerung

$v_{\max}$: Maximalgeschwindigkeit

K_v : Steuerfaktor für die Geschwindigkeit

7.2.5. Steuergrössen

Der Roboter wird einzig über die zwei Antriebsmotoren gesteuert, deren relative Leistungen im Folgenden als f_L (links) und f_R (rechts) bezeichnet werden:

$$f_R = f_v \cdot v \cdot d$$

$$f_L = f_v \cdot v \cdot (1 - d)$$

f_v : Geschwindigkeitsfaktor

v : Geschwindigkeitsvorgabe

d : Steuergrösse für die Richtung (-1 .. 1)

7.2.6. Regelung

Der Reglerentwurf für die Richtung gestaltete sich als relativ schwierig, weil sich der Roboter nicht linear verhält. Als Regler wurde ein generischer PID-Regler eingesetzt, welcher mit Hilfe des Simulink-Modells (Abbildung 39), des Emulators und experimentell parametrisiert wurde. Das grösste Problem ist die begrenzte Bodenhaftung, weshalb direkt vor den Motoren eine Beschleunigungsrampe eingesetzt werden musste. Diese führt sehr schnell zu instabilem Verhalten, was durch die teilweise variierende Zykluszeit des Reglers noch verschärft wird.

Für die Geschwindigkeit wurde komplett auf einen Regler verzichtet, da diese nicht kritisch und nur geringen Störeinflüssen unterworfen ist. Es wird also direkt der Term $f_v \cdot v$ gesteuert.

7.2.7. Echtzeitproblematik

Die Robotersteuerung läuft auf dem SBC in einem normalen Linuxprozess. Dies führt dazu, dass der Scheduler jederzeit auf einen anderen Prozess umschalten kann, was zu wesentlich längeren Durchlaufzeiten der Hauptschleife führt. Ein Thread[5] im Stackoverflow-Forum behandelt ein ähnliches Problem und zeigt die Verwendung der Echtzeit-Prozessattribute in Linux. So kann die Steuerung mit einer viel höheren Priorität arbeiten als der Rest des Systems, was die Zykluszeit deterministischer macht. Es gibt nach wie vor Ausreisser mit einer Verzögerung von ca. 100ms, doch damit kann der Regler in den meisten Fällen gerade noch arbeiten.

Weiterhin wurde versucht, den Linuxkernel mit dem realtime patch (RT-Patch[6]) durch Aktivierung von kernel preemption echtzeitfähig zu machen. Leider stellte sich heraus, dass sich auch die Kernelsourcen des Herstellers des SBCs nicht ohne Weiteres kompilieren lassen. So war nach dem Boot beispielsweise keine USB-Unterstützung mehr vorhanden. Der Support von Technologic Systems hätte mit grosser Wahrscheinlichkeit helfen können, doch auf weitere Untersuchungen in diese Richtung wurde verzichtet.

Eine sauberere Alternative wäre die Verwendung eines Echtzeit-Betriebssystems, bzw. einer Echtzeit-Erweiterung für Linux wie beispielsweise RTAI[7], doch dies würde eine grössere Änderung an der Softwarearchitektur erfordern und wurde daher vorerst auch nicht weiter verfolgt.

[5] http://stackoverflow.com/questions/9374653/real-time-scheduling-in-linux
[6] https://rt.wiki.kernel.org/index.php/Main_Page
[7] https://www.rtai.org/

7.3. Programmablauf

7.3.1. Booten des Betriebssystems

Das Betriebssystem bootet direkt beim Anschliessen des Akkus. Für den Roboter sind drei Startskripts unter /etc/init.d/ abgelegt:

```
/etc/init.d/zstartjingle
/etc/init.d/robot-autostart
/etc/init.d/bt-console
```

Über `zstartjingle` werden zwei kurze Töne ausgegeben, um das Ende des Bootvorgangs (ca. 90s) zu signalisieren.

Der Daemon `robot-autostart` hat die Aufgabe, das Fahrzeug über den Hardware-Knopf zu starten. Ist also der Hardware-Knopf gedrückt, so wird das Programm `/home/eclipse/robot` gestartet.

Mittels `bt-console` wird der Bluetooth-Dongle und die Serial Port Emulation über RFCOMM initialisiert:

```
#bt-consoled
#!/bin/sh
hciconfig 0 reset
hciconfig 0 up
sdptool add SP
nohup rfcomm watch 0 1 > /dev/null &
nohup bt-consoled-loop > /dev/null &
```

Ausserdem sorgt der Daemon `bt-consoled-loop` dafür, dass Terminalverbindungen angenommen werden:

```
#bt-consoled-loop
#!/bin/sh
while [ true ]
do
	while [ ! -c /dev/rfcomm0 ]; do sleep 3; done
	getty -t 3600 115200 /dev/rfcomm0
done
```

7.3.2. Konfiguration & Programmstart

Der Programmeinstiegspunkt befindet sich in der main()-Funktion des robot/main.cpp. Das Hauptprogramm bedient sich hier der Klasse robot::Robot. Diese bietet Methoden zum Initialisieren und Abschalten der Roboterkomponenten sowie eine Routine zur Fahrtsteuerung.

Ausserdem werden im main.cpp folgende Einstellungen geladen und festgelegt:

- abzufahrender Pfad (über Punkte und Bezier-Kurven)
- Geschwindigkeiten auf den einzelnen Pfadabschnitten
- zu erkennende Objekte bei der Bildverarbeitung
- Verhalten an bestimmten Pfadpositionen (über sog. Trigger und Callback-Funktionen)

Abbildung 33 zeigt die aktuellen Parcours-Einstellungen. Der Ursprung des Koordinatensystems liegt beim Endpylon. Vom Fahrzeug aus gesehen zeigt x nach rechts, y nach vorne und z nach oben. Die Fahrzeugposition bezieht sich immer auf die Mitte der hinteren Achse. Die blauen bzw. orangen Bereiche stellen die Suchboxen von Pylonen und Würfel. Die S-Kurve nach dem 2. Tiefenbild darf nicht zu kurz oder zu lang ausfallen, da sich sonst die Reglung aufschaukelt oder Abweichungen in x-Richtung auftreten. Deswegen ist das Ende der S-Kurve frühestens bei 3m, spätestens bei 2.2m, aber optimal 0.1m vor dem Würfel. Bei den Pylonen ist angegeben, welche Breite und Höhe des Querschnitts erwartet wird. Der Würfel ist 50-71mm breit und 50mm hoch.

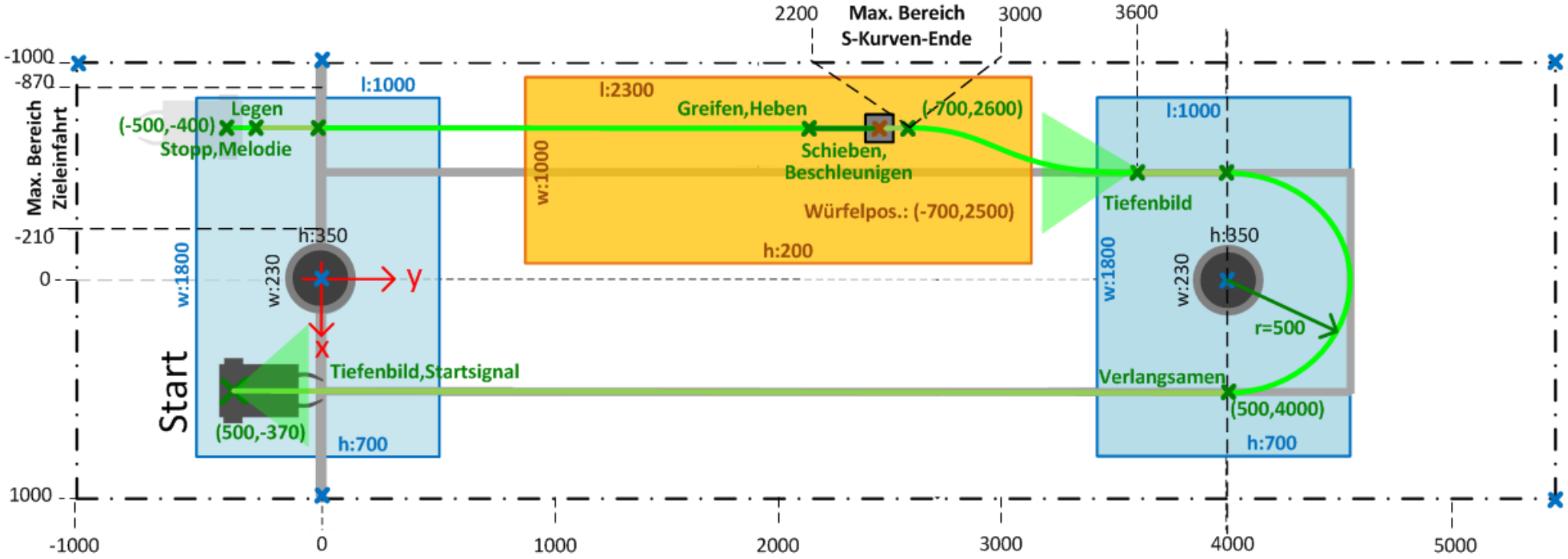


Abbildung 33: Detaillierter Parcours-Ablauf

7.3.3. Winkelkorrektur

Zu Beginn und nach der Kurve wird ein Tiefenbild aufgenommen. Mittels Bildverarbeitung werden darin Pylon und Würfel gesucht. Dank der erwarteten und der realen Position des Kurvenpylons kann festgestellt werden, in welchem Winkel das Fahrzeug zum Parcours steht. Dazu wird die Winkeldifferenz zwischen den zwei Vektoren herangezogen. Nachfolgend die Berechnung am Start:

$$\theta_{Startwinkel\ korrigiert} = \theta_{Startwinkel} + \sin^{-1}\left(\frac{\vec{v}_{FzuPReal} \times \vec{v}_{FzuPErwartet}}{|\vec{v}_{FzuPReal}| * |\vec{v}_{FzuPErwartet}|}\right)$$

wobei $\vec{v}_{FzuPReal} = \left(\vec{p}_{Pylon\ erkannt} - \vec{p}_{Fahrzeug}\right)$,

$\vec{v}_{FzuPErwartet} = \left(\vec{p}_{\ Pylon\ erwartet} - \vec{p}_{Fahrzeug}\right)$,

$\vec{p}_{\ Pylon\ erwartet} = \begin{pmatrix} 4000mm \\ 0 \end{pmatrix}$, $\theta_{Startwinkel} = 90°$ $\vec{p}_{Fahrzeug} = \begin{pmatrix} -370mm \\ 500mm \end{pmatrix}$

Die Würfelposition wird ebenfalls zuverlässig erkannt, bei diesem Schritt aber noch nicht miteinbezogen. Sobald der Winkel korrigiert wurde, ertönt ein Signal und der Start kann mittels Taster ausgelöst werden.

7.3.4. Fahrroutine

Der Kern des Programms ist die Routine zum Fahren und Einhalten eines beliebigen Parcours-Pfades. Diese ist wie folgt implementiert:

```
Zeit seit dem letzten Durchgang messen.
Zurückgelegte Distanzen am rechten und linken Rad anhand der Encoder feststellen.
Zeit und Distanzen dem Modul robot::Navigation übergeben

       robot::Navigation berechnet anhand des festgelegten Pfades
              die aktuelle Fahrzeugposition
              die aktuelle Fahrzeugausrichtung
              Schnittpunkt und Winkel (IST-Wert für PID-Regler)
                     zwischen virtuellem Liniensensor und Pfad
              die neue Fahrzeuggeschwindigkeit
              die neue Richtung anhand des PID-Reglers (SOLL-Wert)
              die Geschwindigkeiten/PWM-Werte der beiden Antriebsmotoren

Neue PWM-Werte den Motoren übergeben

Falls sich das Fahrzeug seit 100 Durchgängen nicht mehr bewegt (d.h. Ziel erreicht)
       Routine beenden
```

7.4. Bildverarbeitung

Das Modul Bildverarbeitung hat die Aufgabe, nach Objekten im Kinect Tiefenbild zu suchen und deren Position zurückzugeben. Die Bildverarbeitung besteht aus folgenden Schritten:

7.4.1. Einlesen des Bildes

Die HAL liefert das zu untersuchende Bild entweder vom darunterliegenden Kinect-Treiber oder der OpenGL-Ausgabe des Emulators. Die Abmessungen betragen 480x680 Pixel. Die Bildpunkte repräsentieren bereits den Tiefenwert an dieser Position in mm. Dieser Wert bewegt sich zwischen 400 und 10000mm.

7.4.2. Festlegen der zu suchenden Objekte

Die Bildverarbeitung kann im Tiefenbild nach mehreren beliebigen Objekten suchen. Diese können wie folgt über die Klasse robot::ObjDef definiert werden:

`position`	Welt-Koordinaten des Objektmittelpunktes
`width`	Breite des Querschnitts
`height`	Höhe des Querschnitts
`area`	Querschnittsfläche
`searchBox`	Kasten im 3D-Raum, wo sich das Objekt befinden kann Definition über Breite, Höhe, Tiefe sowie Mittelpunktkoordinaten

So können die beiden Pylonen und der Würfel definiert werden. Die folgenden Schritte werden für jedes übergebene Objekt vollzogen.

7.4.3. Festlegen der Region of Interest

Damit Performance gespart werden kann, wird der Bildausschnitt berechnet, in dem sich das zu suchende Objekt möglicherweise befindet. Dazu wird die in den Objektdefinitionen festgelegte searchBox herangezogen. Die Eckpunkte dieses Kastens liegen jedoch noch in Weltkoordinaten vor.

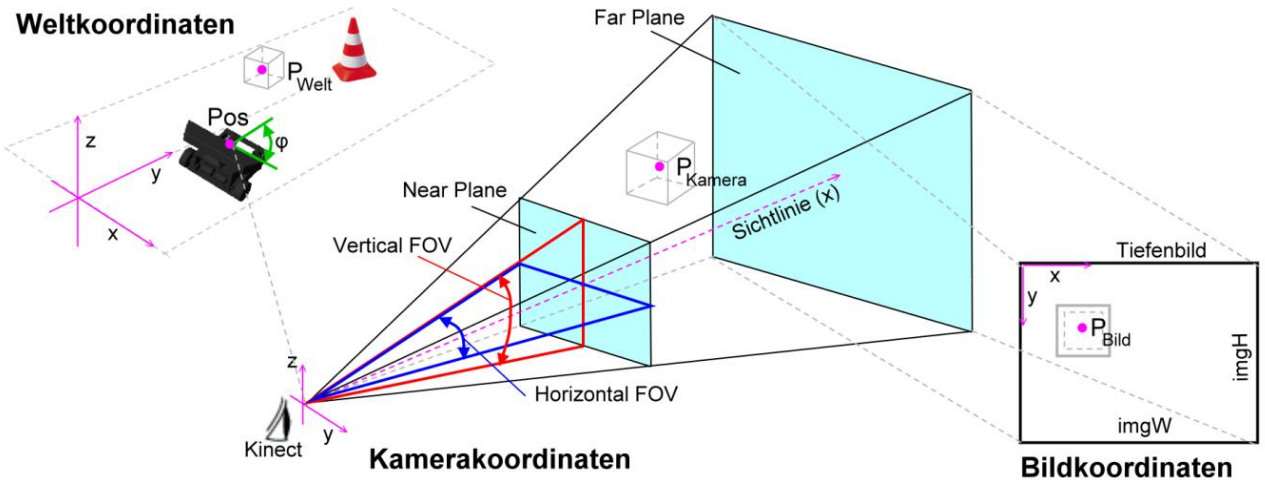


Abbildung 35: Zusammenhand der Koordinatensysteme

Die Klasse utils::Projector bietet Methoden an, um Punkte auf die Bildebene zu projizieren. Der Punkt muss zuerst in Kamerakoordinaten umgewandelt werden, was mit einer Translation zur Kameraposition Pos und einer Rotation um den Kamerawinkel φ möglich ist. Die Matrix-Multiplikation sieht also wie folgt aus:

$$P_{Kamera} = \begin{pmatrix} \cos(-\varphi) & -\sin(-\varphi) & 0 & -Pos_x \\ \sin(-\varphi) & \cos(-\varphi) & 0 & -Pos_y \\ 0 & 0 & 0 & -Pos_Z \\ 0 & 0 & 0 & 1 \end{pmatrix} * P_{Welt}$$

Um die Kamerakoordinaten in P_{Kamera} Bildkoordinaten P_{Bild} zu überführen, folgt nun eine perspektivische Projektion. Anhand des Frustums sieht man, dass zur Berechnung von horizontalen und vertikalen Distanzen im Bild folgender Faktor angewendet werden kann:

$$xFact = \frac{Bildbreite}{2 * \tan(horizontal\ FOV)} \qquad yFact = \frac{Bildhöhe}{2 * \tan(vertical\ FOV)}$$

$horizontal\ FOV$: horizontaler Bildwinkel in rad, 58° für Kinect

$vertical\ FOV$: vertikales Bildfeld in rad, 45° für Kinect

Zur Berechnung des Bildpunktes fehlt nur noch eine Koordinatentransformation auf die obere linke Bildecke.

$$P_{Bild} = \begin{pmatrix} \frac{-P_{Kamera_Y} * xFact}{P_{Kamera_X}} + \frac{Bildbreite}{2} \\ \frac{-P_{Kamera_Z} * yFact}{P_{Kamera_X}} + \frac{Bildhöhe}{2} \\ P_{Kamera_X} \end{pmatrix}$$

Einfachheitshalber wird nun die Region of Interest auf die äussersten Eckpunkte der transformierten SearchBox festgelegt. Bei der weiteren Bildverarbeitung werden nur die Pixel berücksichtigt, die sich innerhalb dieses Bildausschnitts und innerhalb Minimal- und Maximaltiefe befinden.

7.4.4. Connected Component Labeling

Im zu untersuchenden Bildbereich werden nun zusammengehörige Flecken bzw. Regionen gesucht. Dies erfolgt über ein Connected Component Labeling (CCL). Dabei werden die Pixel mit ihren Nachbarn verglichen und mit Labels (Nummern) markiert. Gehören die Pixel zusammen, d.h. überschreitet die Differenz der Tiefenwerte einen definierten Schwellwert nicht, so erhalten die Pixel dasselbe Label. Der Algorithmus besteht aus folgenden Schritten

```
Iteriere zeilenweise durch den zu untersuchenden Bildausschnitt
	Nachbaren suchen
		West=links, Nordwest=links oben
		Nord=oben, Nordost=rechts oben
	Gehören West und aktuelles Pixel zusammen?
		Dem aktuellen Pixel das Label von West zuweisen
		Gehören Nordwest und West zusammen und haben unterschiedliche Labels?
			Labels von Nordwest und West zusammenführen
		Gehören Nord und Nordwest zusammen und haben unterschiedliche Labels?
			Labels von Nord und Nordwest zusammenführen
		Gehören Nordost und Nord zusammen und haben unterschiedliche Labels?
			Labels von Nordost und Nord zusammenführen
	Sonst, gehören Nordwest und aktuelles Pixel zusammen?
		Dem aktuellen Pixel das Label von Nordwest zuweisen
		Gehören Nord und Nordost zur selben Region und haben unterschiedliche Labels?
			Labels von Nordost und Nord zusammenführen
	Sonst, gehören Nordwest und aktuelles Pixel zusammen?
		Dem aktuellen Pixel das Label von Nordwest zuweisen
	Sonst
		Dem aktuellen Pixel ein neues Label zuweisen
Zusammengeführte Labels bilden eine Region
Pro Region
	Fläche, Koordinate oben links und unten rechts speichern
```

Die Zusammengehörigkeit von Labels wird in einer doppelt verketteten Liste gespeichert. Zwei Labels bzw. Labelgruppen werden zusammengeführt, indem Ende und Anfang verknüpft werden. In Abbildung 36 wird der Algorithmus abgebildet. Das rote Kästchen ist das aktuelle Pixel und erhält das Label „1“ seines linken Nachbars. Ausserdem wird festgestellt, dass der Nachbar oben zur selben Region gehört.

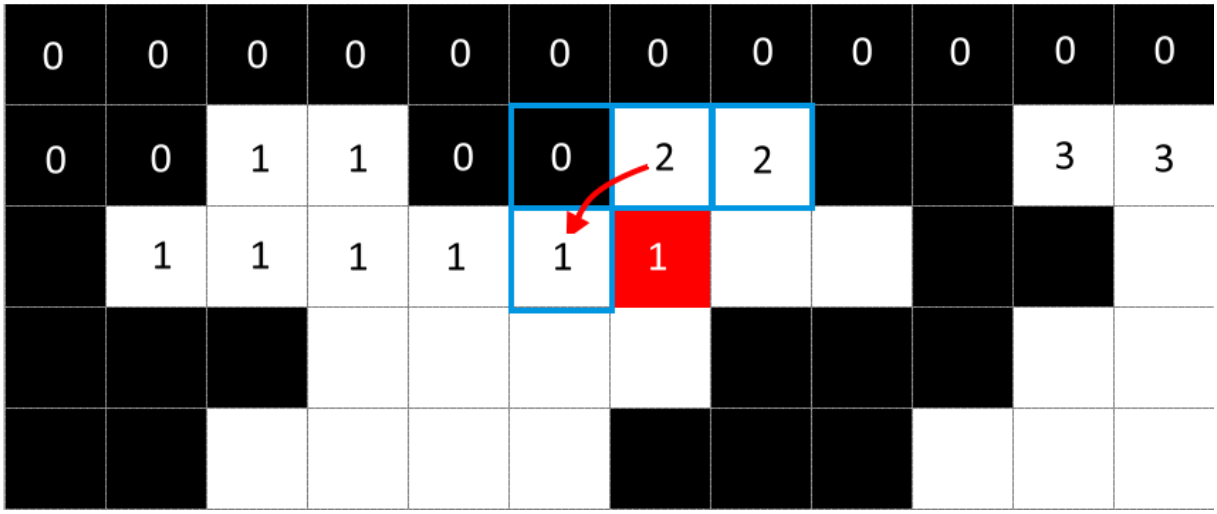


Abbildung 36: Connected Component Labeling

7.4.5. Filtern nach ungültigen Kandidaten (Fläche, Zentrum)

Aus den erhaltenen Regionen bzw. Kandidaten werden nun Treffer gesucht. Dazu wird zuerst das geometrische Zentrum berechnet (Mittelwert aller x- und y-Positionen der zugehörigen Pixel):

```
int sumX = 0;
int sumY = 0;
int count = 0;
for(int y = y0; y < y1; y++) {
  for(int x = x0; x < x1; x++) {
    if(cclData[y*IMGRECOGNITION_WIDTH+x]>0) {
      sumX += x;
      sumY += y;
      count++;
    }
  }
}
int cx = sumX / count;
int cy = sumY / count;
```

Der Mittelpunkt wird schliesslich über den Projector in Weltkoordinaten zurücktransformiert.

Liegt das Zentrum nicht in der definierten Suchbox, fallt der Kandidat weg.

7.4.6. Suchen des besten Treffers

Bei genau einem Treffer wird keine Bewertung vorgenommen. Wenn mehrere Treffer vorhanden sind, wird mit folgenden Kriterien eine Bewertung vorgenommen:

Position	Minimale Distanz zwischen Objektposition und Suchbox-Mittelpunkt
Form	Differenz erwartete Form und gemessene Form Beim Würfel: Vergleich effektive Fläche und Fläche d. Bounding Box Beim Pylon: Vergleich effektive Fläche und Fläche eines Kegelquerschnitts
Proportionen	Differenz erwartetes und gemessenes Breiten-Höhen-Verhältnis
Fläche	Differenz erwartete und gemessene Querschnittsfläche

Als Ergebnis wird der Treffer mit der besten Wertung gespeichert.

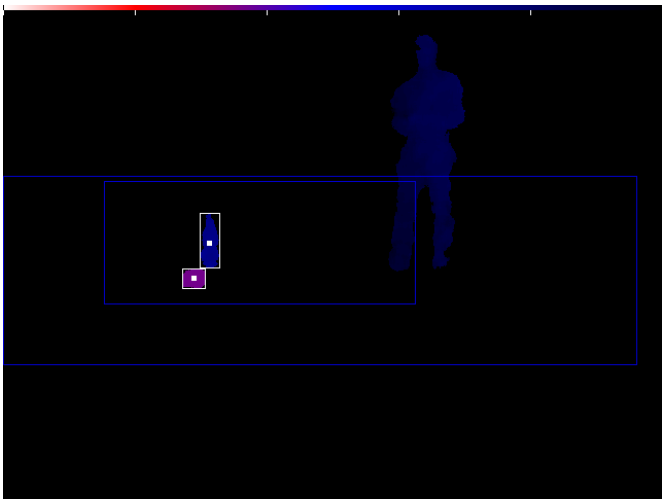


Abbildung 37: Kinect-Tiefenbild mit Debug-Infos

Abbildung 37: Die Skala oberen Rand des Debug-Bildes zeigt die Tiefe in Metern an. Das Bild wurde nach der Kurve aufgenommen, wobei der Würfel recht nahe Endpylon und in der Ecke des möglichen Bereichs liegt. Die blauen Rechtecke markieren den Bereich, in dem Würfel (grosses Rechteck) und Pylon (kleines Rechteck) erwartet werden. Die weissen Rechtecke umranden die gefundenen Objekte

7.5. Emulator

Zur Unterstützung der Softwareentwicklung wurde bereits in PREN1 mit dem Schreiben eines Hardwareemulators begonnen. Dieser sollte es ermöglichen, auch ohne Zugriff auf den Roboter an der Steuerungssoftware zu arbeiten. Die zentralen Aufgaben, welche der Emulator erfüllen muss, sind:

- Modellierung des physischen Verhaltens des Fahrzeuges bei gegebenen Steuersignalen zu den Motoren
- Generierung eines Datenstroms mit Tiefenwerten, wie auch die Kinect erzeugt wird
- Volle Kompatibilität zur Robotersteuerung, so dass möglichst keine Anweisungen zur bedingten Kompilierung im Code benötigt werden.

Gerade die Programmierung eines möglichst guten physikalischen Modells des Roboters ist eine relativ komplexe Aufgabe. Daher wurden zur Vereinfachung einige Annahmen getroffen:

- Der Roboter wird mit homogener Masseverteilung modelliert
- Der Schwerpunkt liegt in der Mitte zwischen den beiden Antriebsrädern
- Die Räder übertragen das Drehmoment ohne Schlupf auf den Boden

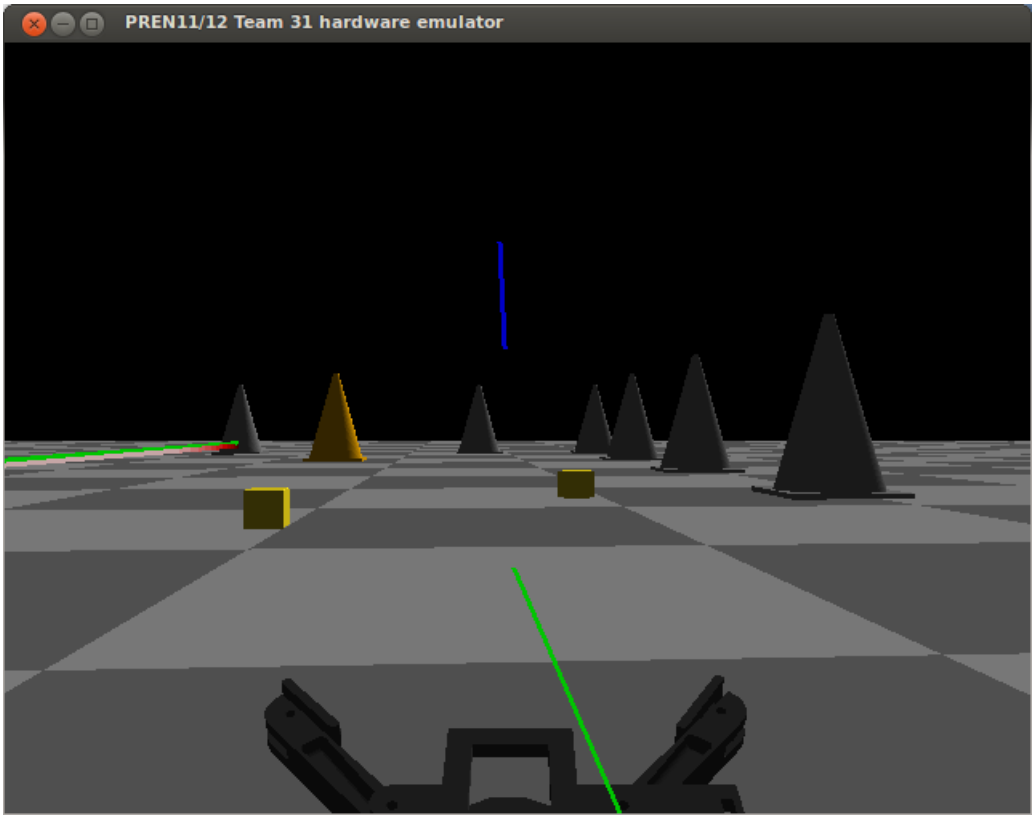


Abbildung 38: Screenshot Emulator

Mit diesen Vereinfachungen war es möglich, das Blockdiagramm auf der nächsten Seite zu erstellen.

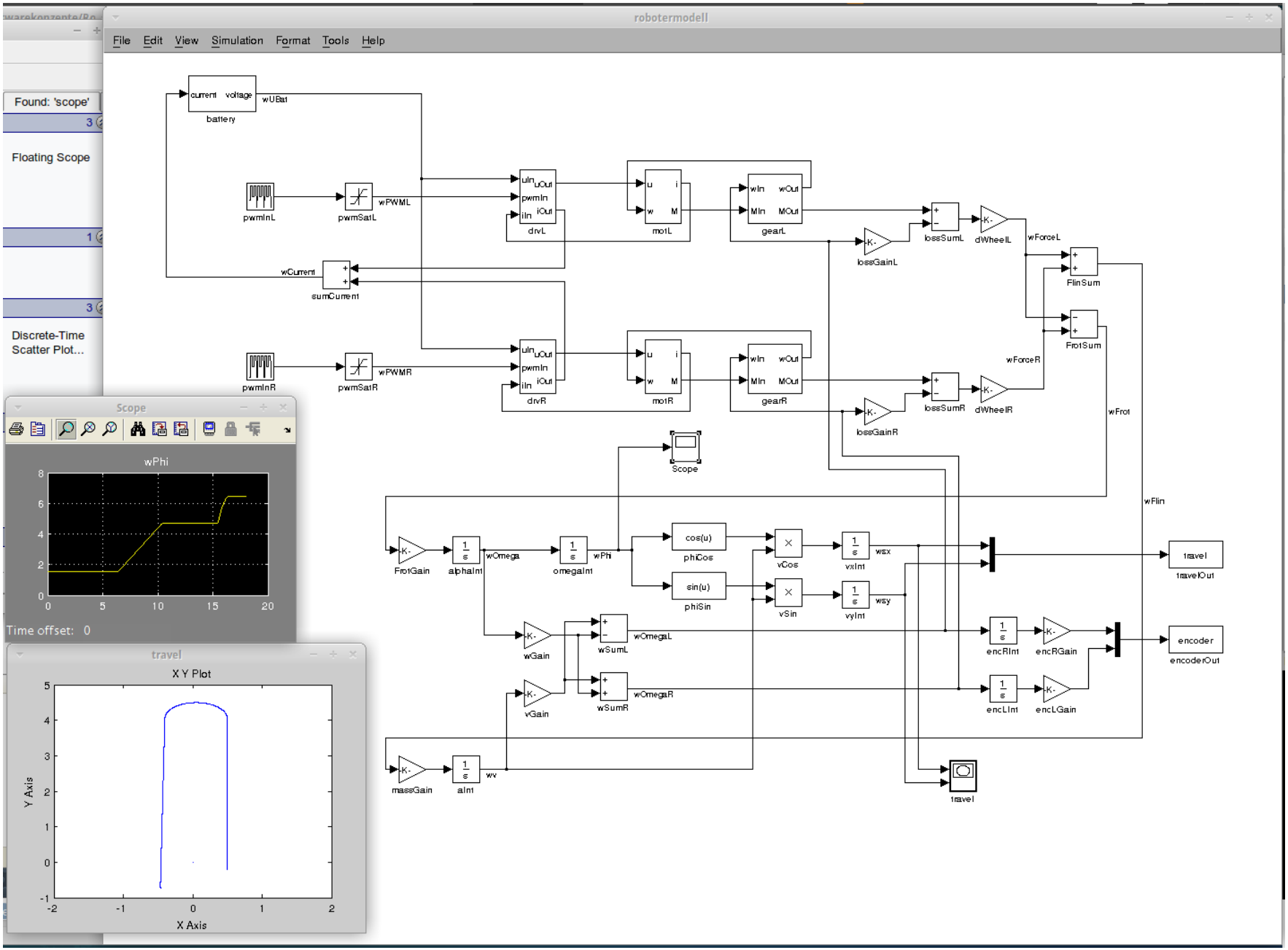


Abbildung 39: Blockdiagramm des Roboters in Simulink

8. Elektronik

8.1. Blockschaltbild der gesamten Elektronik

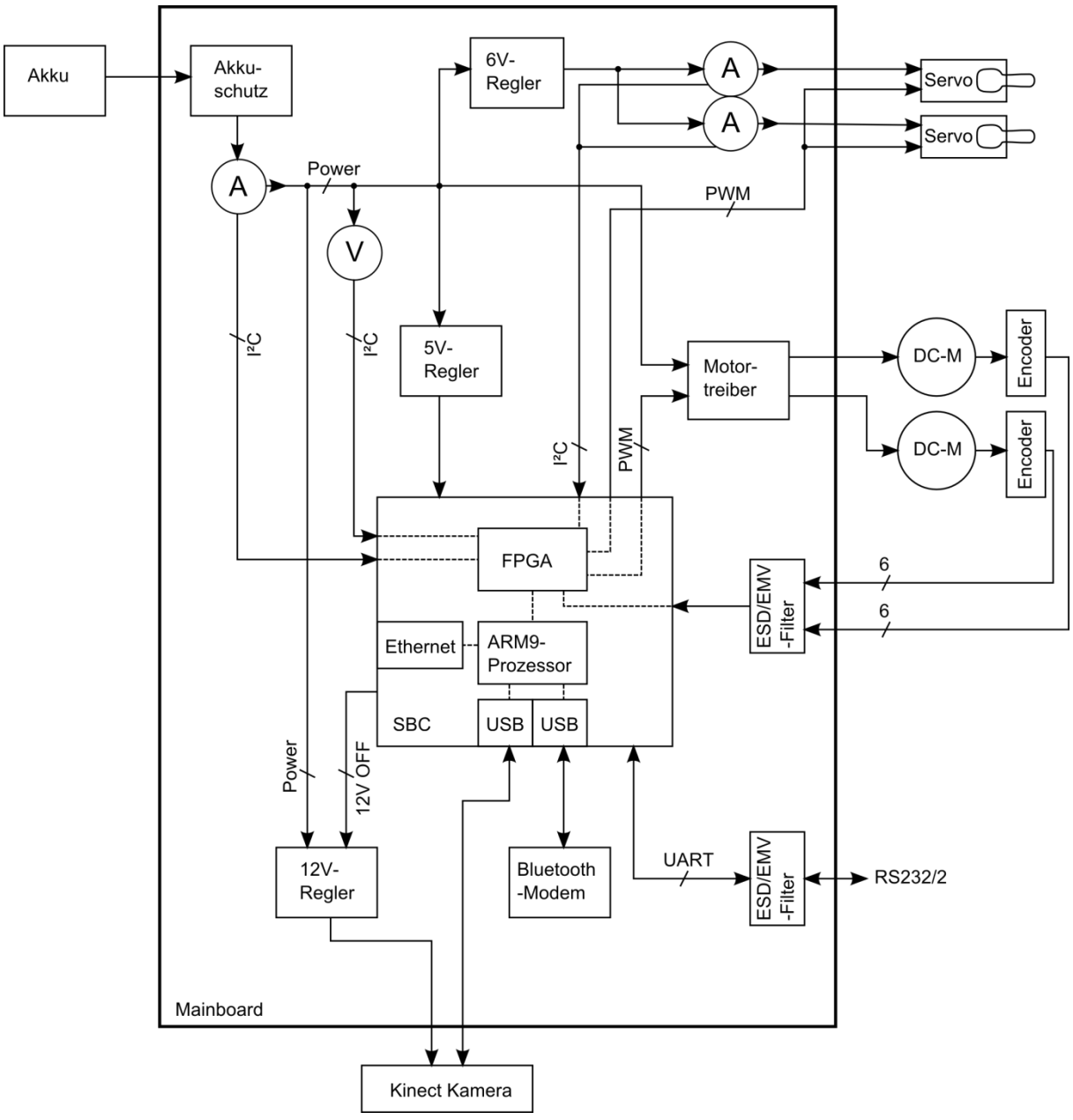


Abbildung 41: Blockschema der Elektronik

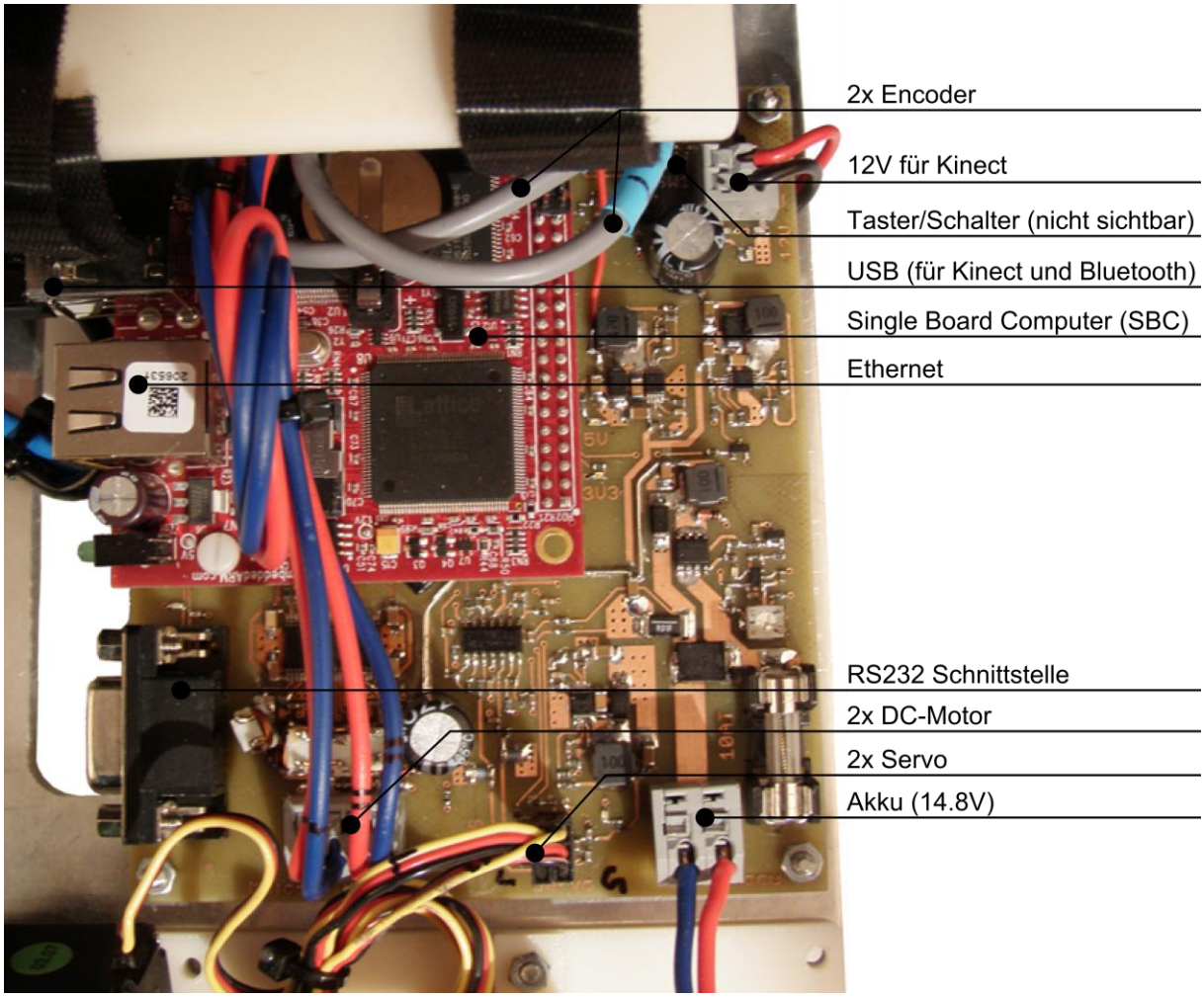
Das Blockschema zeigt die Hauptplatine des Fahrzeugs. Das Kernstück ist ein Single-Board-Computer (SBC) mit einem ARM9-Prozessor, einem FPGA und Schnittstellen wie Ethernet und USB. Das FPGA eignet sich für zyklische Aufgaben, welche für den Prozessor zu viel Rechenzeit in Anspruch nehmen würden, wie beispielsweise I^2C-Interface, PWM-Generator und Encoder-Auswertung. Der Prozessor übernimmt die Aufgaben für die Regelung der Strecke und die Auswertung der Bilder, welche von der Kinect Tiefenkamera ausgegeben werden.

Für die Spannungsversorgung wird ein Modellbau-Akku Pack mit 4 LiPo-Zellen (14.8V) verwendet, aus welchem direkt die Motortreiber versorgt werden. Ein 5V-Spannungsregler stellt eine stabile 5V-Versorgung für den SBC zur Verfügung. Ein 6V-Spannungsregler speist die beiden Servos und der 12V-Regler wird für die Kinect verwendet.

Als Antriebsmotoren dienen zwei DC-Motoren an den beiden Hinterrädern. An beiden Rädern sitzt jeweils ein magnetischer Encoder mit einer Auflösung von 10Bit. Die Signale dieser Encoder werden im FPGA ausgewertet und für die Regelung verwendet.

Die Konfiguration und Steuerung des Fahrzeugs kann entweder kabelgebunden über Ethernet/ RS232 oder kabellos über Bluetooth vorgenommen werden.

8.2. Anschlüsse der Hauptplatine



8.3. Funktion der Schaltung

8.3.1. Eingangsteil

Die Hauptaufgabe des Eingangsteils ist es, bei zu niedriger Akkuspannung den Akku elektrisch von der Schaltung zu trennen. Für die Abschaltspannung wurde 12V gewählt, dies ergibt 3V pro Zelle bei einem 4-Zellen-Lipo-Akku. Als Spannungsreferenz dient ein LM431 mit einer Referenzspannung von 2.5V. Mit einem Potentiometer kann die Ausschaltspannung genau abgestimmt werden. Es handelt sich dabei um das einzige Potentiometer in der gesamten Schaltung.

Die Hysterese von etwa 0.5V, welche aus den zwei Widerständen R6 und R9 realisiert ist, soll Oszillationen des Komparators unterdrücken.

Die zwei LEDs signalisieren die Funktion der Schaltung. Bei „rot“ liegt die Akkuspannung nicht an der Schaltung, T1 sperrt also. Bei „grün“ ist die Akkuspannung höher als 12V, T1 leitet. L1, C7 und C8 dienen zur Filterung von hochfrequenten Störungen auf der Speisespannung Vin. Die Schottky-Diode D3 agiert als Freilaufdiode von L1 beim Abschalten des FETs.

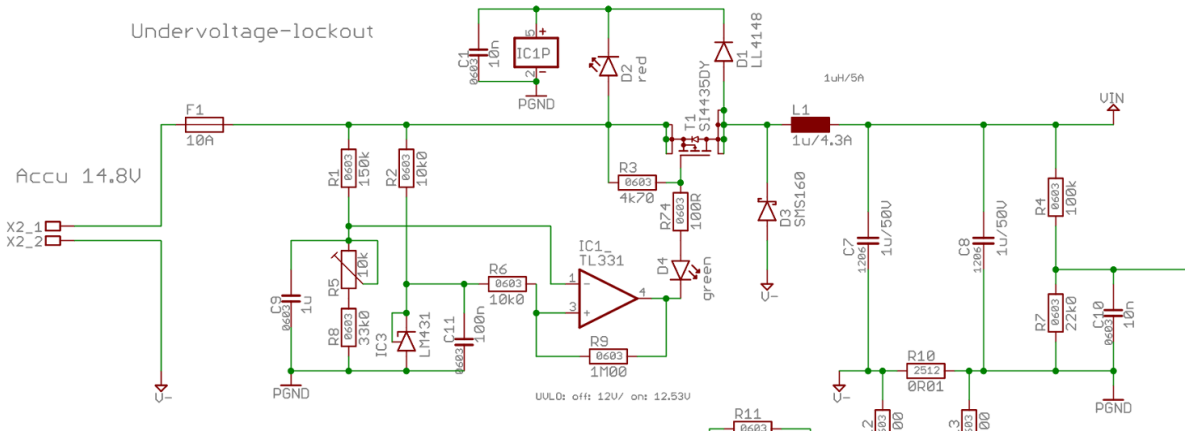


Abbildung 42: Unterspannungs-Abschaltung (undervoltage lockout)

8.3.2. Spannungs- und Strommessung

Der Prozessor soll die Akkuspannung auch messen können, damit der Benutzer nicht von einer plötzlichen Abschaltung überrascht wird. Bei niedrigem Akkustand soll über den eingebauten Piezo-Summer ein Warnsignal ausgegeben werden können. Dazu wird mit dem AD-Wandler (AD7995) über einen Spannungsteiler die Akkuspannung (ADC_V) gemessen, zusätzlich wird über einen 10mΩ Shunt-Widerstand die gesamte Stromaufnahme (ADC_I_TOT) der Schaltung gemessen.

8.4. Spannungsregler

8.4.1. Getaktete Speisung

Getaktete Speisung wurde überall dort eingesetzt, wo etwas höhere Ströme, 0.5A bis 1.5A benötigt werden. Die drei Hauptverbraucher sind die Kinect Kamera (12V-Speisung), der SBC (5V) und die Servos (6V). Bei allen drei Speisespannungen regelt jeweils ein TPS54140, ein Step-Down-Converter, der bis 42V und 1.5A arbeitet.

Die Bauteile rings um die Regler sind identisch, bis auf einige für die jeweilige Speisspannung individuellen Werte. Die Speisung für die Kinect erhielt zudem einen zusätzlichen Transistor (T2) um den Spannungsregler bei Nichtbenutzung der Kinect abschalten zu können.

Der EN-Eingang (enable) braucht eine gewisse Spannung, damit der Regler überhaupt arbeiten kann. Diese Unterspannungs-Abschaltung (undervoltage lockout) kann mit dem Spannungsteiler eingestellt werden. Am SS/TR-Pin bestimmt der Kondensator die Ausgangsanstiegszeit (output rise time). Der Widerstand am RT/CLK-Pin gibt die Taktfrequenz der Speisung vor. Die Bauteile am COMP-Pin dienen als Kompensations-Netzwerk. Die Werte müssen so gewählt sein, dass der Regler nicht schwingt. Am Schaltausgang PH (Ausgangsseite) dienen eine Speicherdrossel, ein 10uF-Kondensator und eine Schottky-Diode zur Vervollständigung des Tiefsetzstellers (Buck-Converter). Die Ausgangsspannung wird mit dem Spannungsteiler am VSENSE-Pin eingestellt.

Alle drei getakteten Regler signalisieren mit einer LED, dass sie eingeschaltet sind.

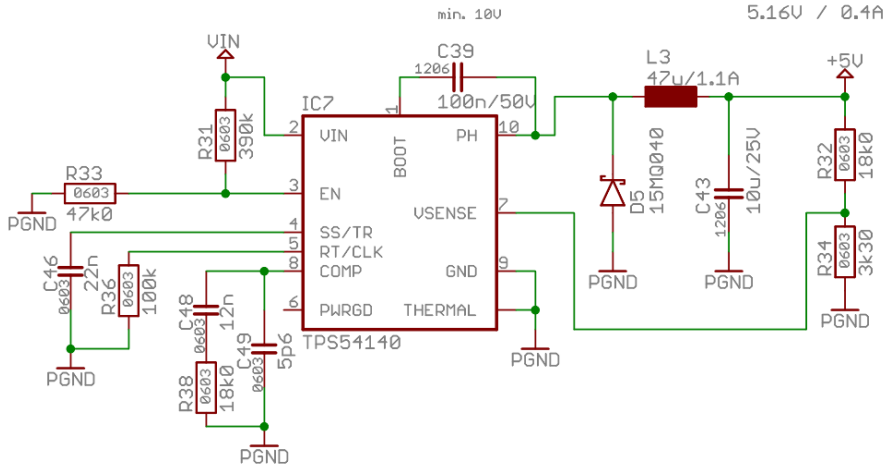


Abbildung 43: 5V-Speisung für den SBC

8.4.2. Linearregler

Die Logik des Motortreibers benötigt eine Spannung von 12V. Diese Spannung wird mit einem Linearregler (LP2951) erzeugt. Dieser Regler benötigt ausser einem Spannungsteiler zur Einstellung der Ausgangsspannung und Stützkondensatoren keine weitere Peripherie.

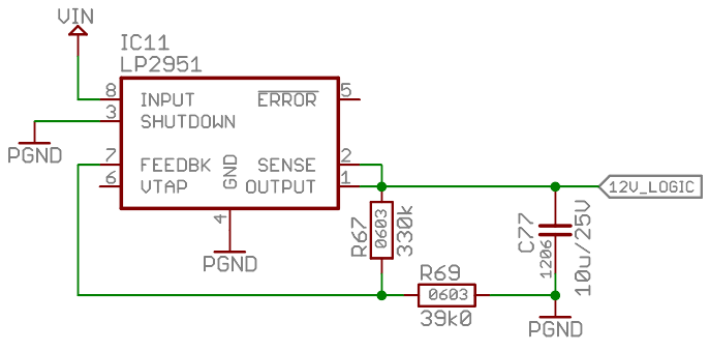


Abbildung 44: 12V-Speisung für die Logik des DRV8412

8.5. Motorentreiber

Der Motorentreiber (DRV8412) benötigt einige zusätzliche Bauteile, in erster Linie Stützkondensatoren. An den Treiber-Ausgängen befinden sich im aktuellen Schema noch 0Ω Widerstände in Serie zu 1nF-Kondensatoren. Dies darf so aber nicht bestückt werden, weil die Kondensatoren AC-mässig, also bei jedem Schalten der internen Leistungstransistoren, einen Kurzschluss gegen Masse darstellen. Weil die Kondensatoren an den Klemmen bestückt sind, müssen die Widerstände durch Spulen ersetzt werden, eingelötet sind nun 2uH-Spulen.

Der DRV8412 besitzt vier Halbbrücken oder zwei Vollbrücken, welche mit vier PWM-Eingängen angesteuert werden. Die zwei Reset-Eingänge (RESET_AB und RESET_CD, beide active-low) dienen zur Abschaltung der jeweiligen H-Brücken. Weitere Features des DRV8412 sind die Unterspannungs-, Überstrom- und Übertemperatur-Abschaltung, welche mit zwei Ausgängen (OTW und FAULT) signalisiert werden. Als Übertemperatur gilt typischerweise ein Wert von 125°C.

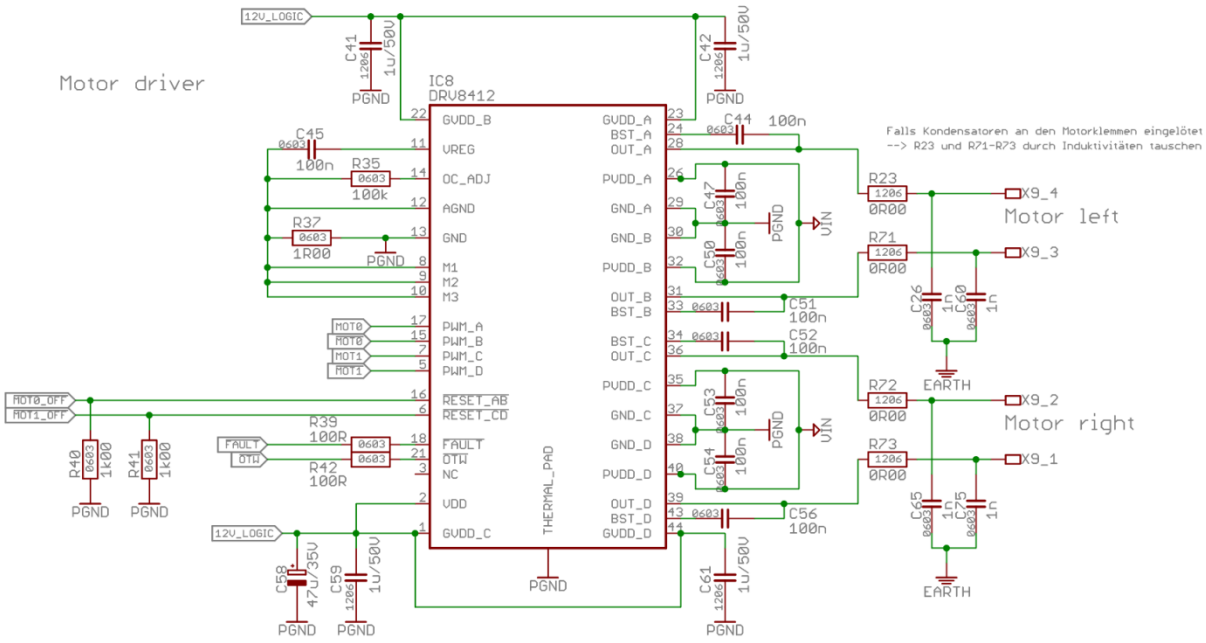


Abbildung 45: Beschalteter Motorentreiben DRV8412

8.6. Encoder

Um der Steuerung eine schnelle Rückmeldung über die Bewegung des Roboters zu geben, wurde an beiden Radachsen je ein Encoder angebracht. Anders als ursprünglich geplant kommen dabei keine selbst gebauten optischen Encoder zum Einsatz, sondern integrierte Bausteine mit einem magnetischen Messprinzip. Es handelt sich dabei um Encoder des Typs AS5040[8] vom Hersteller Austria Microsystems (AMS).

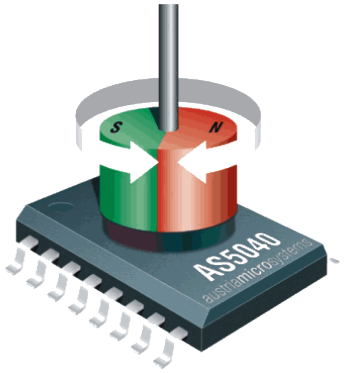


Abbildung 46: Encoderbaustein AS5040

Der Winkel des zylindrischen und diametral magnetisierten Magnets wird mit 10 Bit (1024 Schritte) aufgelöst und als Quadratursignal ausgegeben. Bei einem Raddurchmesser von ca. 65mm ergibt dies eine Wegauflösung von 0.2mm.

Den Ausschlag zur Verwendung dieses Bauelements gab die grosse Auflösung, welche mit einer selbst gefertigten Lochscheibe kaum möglich gewesen wäre. Ausserdem ist der AS5040 mit einem Preis von $6.51 verglichen mit optischen Encodern sehr günstig und konnte direkt bei AMS als kostenloses Muster bezogen werden.

[8] http://www.ams.com/eng/content/download/1285/7214

8.7. Strommessung der Servos

8.7.1. Beschreibung

Wenn das Fahrzeug den Würfel ergreift, soll es erkennen können, ob es den Würfel in der Zange hält oder ob es danebengegriffen hat. Falls sich der Würfel in der Zange befindet, ist der Servostrom höher. Wenn er sich nicht in der Zange befindet, befindet sich das Servo im „Leerlauf“, folglich ist der Strom kleiner.

Die Strommessung wurde gleich für beide Servos realisiert, um alle vier Operationsverstärker des TLV2374 und alle ADC-Kanäle zu benützen. Die Minus-Anschlüsse der Servos laufen anstatt direkt auf Masse zuerst über jeweils einen 50mΩ Shunt-Widerstand. Es wird die am Shunt (im Bild R50) abfallende Spannung gemessen, verstärkt und so als Strom interpretiert.

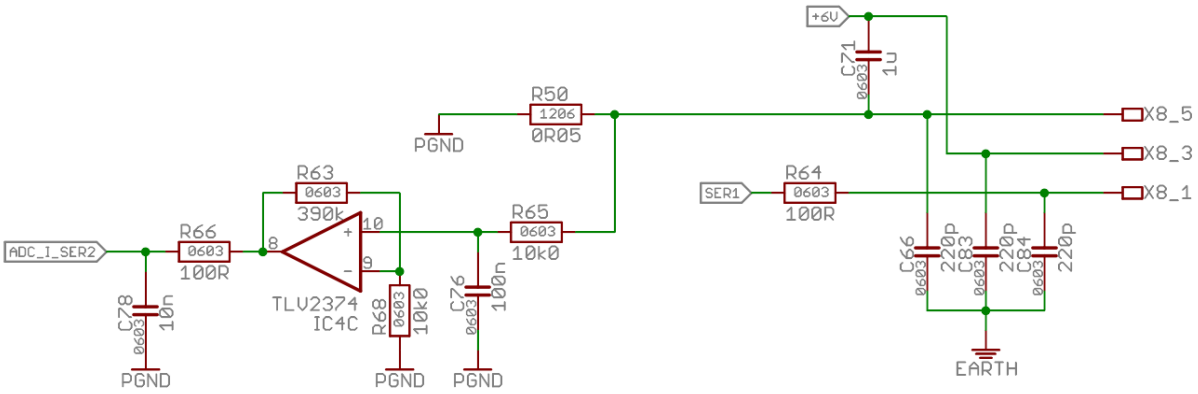


Abbildung 47: Strommessung eines Servos

8.7.2. Messbereich

Die Verstärkung des nicht inv. Verstärkers beträgt: $V = \frac{390k\Omega}{10k\Omega} + 1 = 40$

Mit einem Range von 0V bis 3.3V des ADC’s ergibt sich für die Shunt-Spannung:

$$U_{Shunt} = \frac{3.3V}{40} = 82.5mV$$

und für den Servostrom: $I_{Servo} = \frac{82.5mV}{50m\Omega}$ =1.65A

Der Servostrom kann also von 0A bis 1.65A (pro Servo) gemessen werden.

8.8. Optionale Baugruppen

8.8.1. RS232-Interface

Eine weitere Kommunikationsmöglichkeit bietet die RS232-Schnittstelle. Ein geeignetes IC (ICL3232) wird hierfür verwendet.

8.8.2. Piezo-Summer

Manchmal möchte man wichtige Hinweise akustisch signalisieren, sodass diese gleich wahrgenommen werden. Der Piezo-Summer kann beispielsweise niedrige Akkuspannung anzeigen.

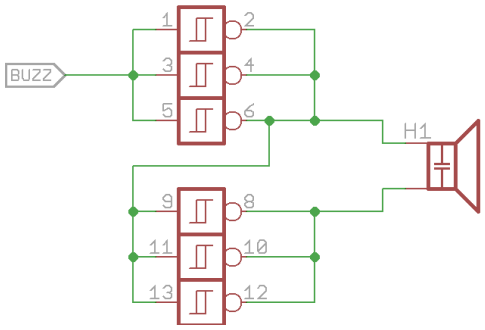


Abbildung 48: Ansteuerung des Piezo-Summers

In der ursprünglichen Schaltung war der Summer relativ leise, da er ein Rechteck-Signal von nur 0/3.3V erhält. Die Schaltung wurde im Nachhinein noch mit einem weiteren Schmitt-Trigger versehen, welcher mit 5V versorgt wird. So erhält der Summer ein Signal von -5V/+5V und tönt dementsprechend auch lauter.

8.8.3. Tasteranschlüsse

Mit den optionalen Taster-/Schalteranschlüssen kann zum Beispiel mit einem Taster das Startsignal an das Gerät gegeben werden.

8.9. EMV-Massnahmen

Als Massnahmen für gute EMV wurden an jeder Buchse/Klemme Kondensatoren im Wert von einigen pico-Farad angelötet. Diese Kondensatoren sollen HF-Störungen, welche in die Schaltung ein- oder aus der Schaltung austreten, gegen eine Filter-Masse (EARTH) kurzschliessen. Diese Filter-Masse ist mit einem 0Ω-Widerstand mit Masse verbunden, damit sie DC-mässig auf definiertem Potential liegt.

Weiter wurde versucht, wichtige Punkte der EMV beim Layout zu beachten:

- Es sollten keine höheren Ströme nahe empfindlicher Elektronik (Prozessoren, Analogteil,…) fliessen.
- Es soll eine separate Filter-Masse vorhanden sein.
- Leiterbahnen sollten immer direkt auf Stützkondensatoren geführt sein, und von diesen dann wieder weg, damit die Stützwirkung am besten ist.

8.10. FPGA

8.10.1. Übersicht

Neben dem ARM9-Prozessor ist auf dem SBC auch ein FPGA verbaut. In der Standardkonfiguration kümmert sich dieses um die Verwaltung einiger Systemkomponenten wie SD-Speicherkarte, RTC, serielle Schnittstelle, etc. Im Rahmen des PREN wurde diese Konfiguration um weitere Komponenten ergänzt um den Prozessor zu entlasten.

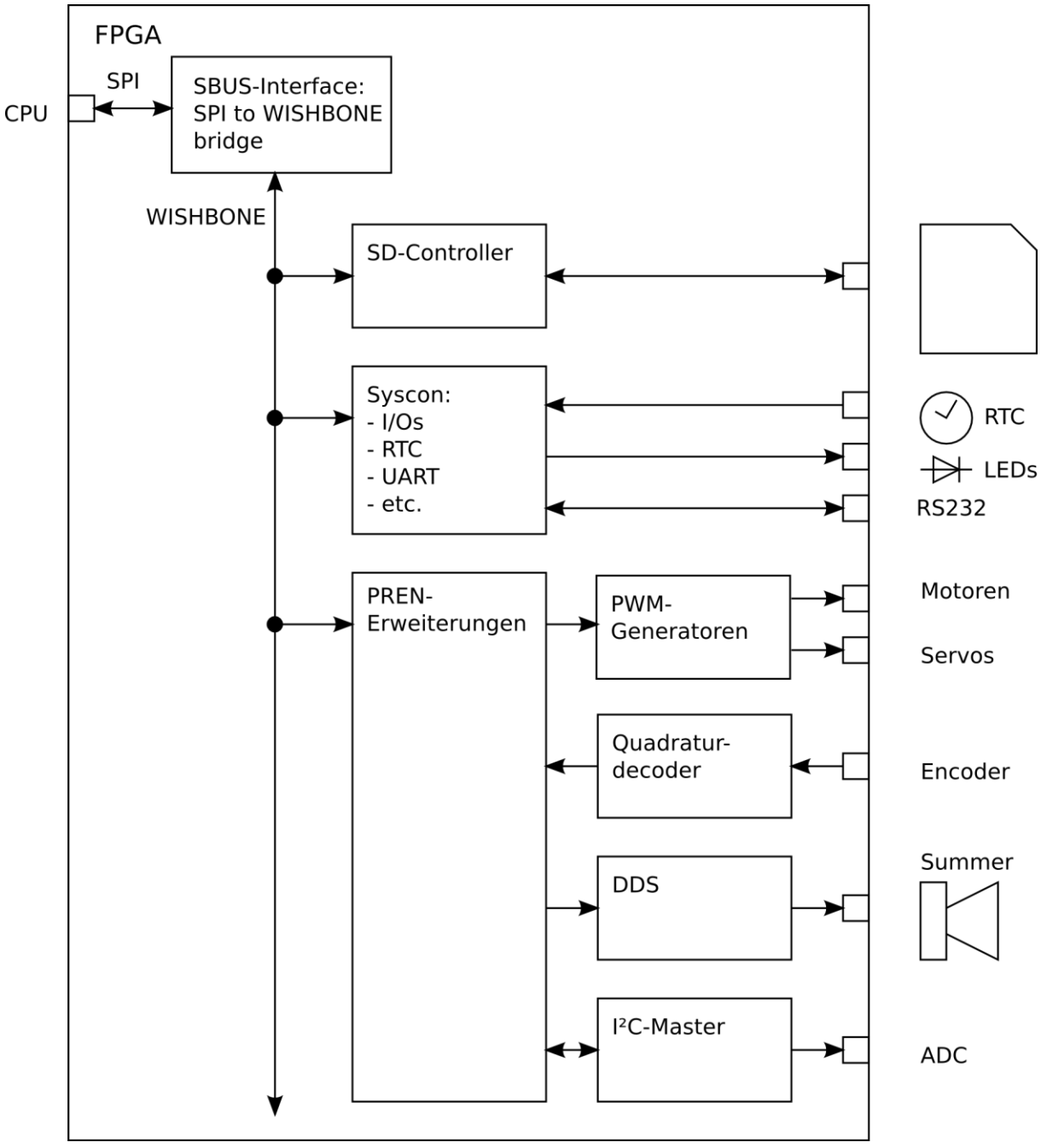


Abbildung 49: FPGA-Übersicht

8.10.2. PWM-Generatoren

Sowohl die Antriebsmotoren wie auch die Servos für den Greifer werden mit PWM-Signalen gesteuert. Dafür wurden je zwei PWM-Kanäle mit 25kHz und 10 Bit Auflösung für die Motoren und 50Hz bei 16 Bit Auflösung für die Servos implementiert.

8.10.3. Quadraturdecoder

Zur Erfassung des gefahrenen Weges sind an beiden Antriebsrädern des Fahrzeugs magnetische Encoder angebracht. Die erzeugten Quadratursignale werden im FPGA dekodiert und in einem vorzeichenbehafteten 16 Bit-Zählregister akkumuliert.

8.10.4. DDS

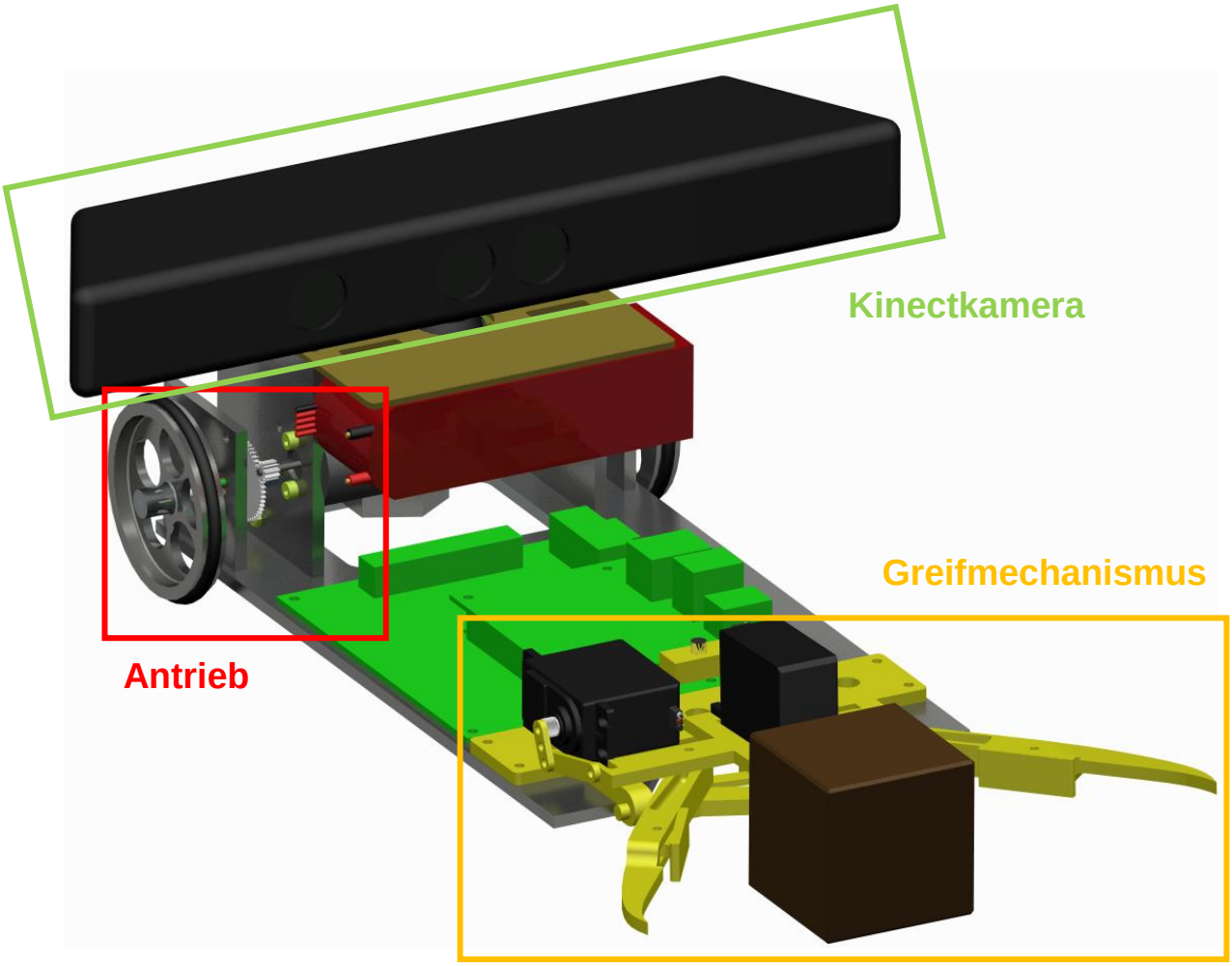
Für die Ansteuerung des Piezo-Summers ist in der FPGA-Konfiguration ein DDS-Block implementiert. Damit lässt sich ein Rechtecksignal im Bereich von 0Hz bis 20kHz in 1Hz-Schritten generieren.

8.10.5. I²C-Master

Der weiter oben erwähnte Analog-Digital-Konverter (ADC) AD7995 besitzt eine I²C-Schnittstelle zur Konfiguration und zum Auslesen der Messwerte. Der Prozessor enthält bereits ein I²C-Interface, doch die regelmässigen Transaktionen würden die Hauptanwendung dauernd unterbrechen und dadurch eine unnötig hohe Prozessorlast verursachen. Aus diesem Grund wurde die FPGA-Konfiguration noch um einen einfachen I²C-Master erweitert, der die Messwerte des ADCs zyklisch ausliest, den Mittelwert über einige Samples bildet und diesen der CPU übergibt.

9. Mechanik

9.1. Übersicht



Das Fahrzeug besteht im Wesentlichen aus drei Teilbereichen. Dies sind erstens die beiden Antriebe, je einer links und rechts, welche für die Fortbewegung und Steuerung verantwortlich sind. An zweiter Stelle ist der Greifmechanismus, welcher für die Ergreifung und das Anheben des Würfels zuständig ist. Als Drittes ist die Kinectkamera mit ihrer Befestigung. Sie ist für die Erkennung der Pylone und des Würfels verantwortlich.

9.2. Antrieb und Fahrgestell

Das Getriebe, zur Übersetzung der Motorenleistung auf das Antriebsrad, konnte man auf Grund der gegebenen Motorendrehzahl und der gewünschten Fahrzeuggeschwindigkeit entsprechend auslegen. Somit wurde eine Übersetzung von 1:25 errechnet. Dies hatte zur Folge, dass die Getriebe zweistufig werden mussten. Aus dem Sortiment der Firma Conrad erhält man die passenden Stirnräder, wodurch ein Getriebe mit der ersten Stufe von 10 auf 50 Zähnen und der zweiten Stufe von 12 auf 60 Zähnen entstand. Der Modul beträgt 0.5mm. Das letzte Stirnrad wurde zusätzlich über Stifte mit einem Stellring verbunden, damit das Stirnrad auf der Welle das volle Drehmoment problemlos übertragen kann.

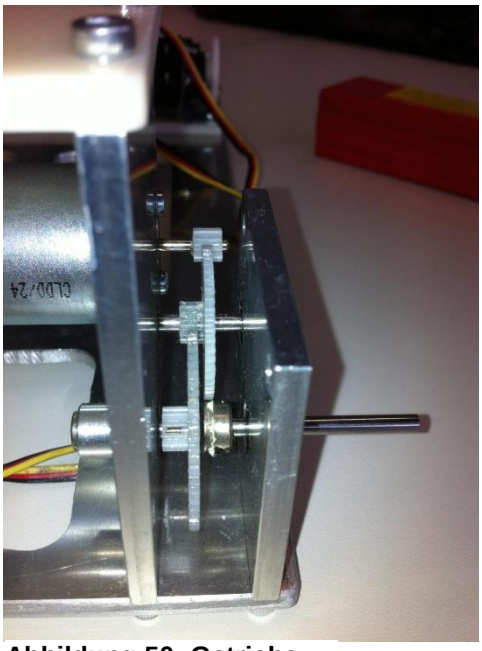


Abbildung 50: Getriebe

Das Rad wird über einen Gewindestift auf die Antriebsachse montiert. Am anderen Ende der Antriebsachse wird ein Magnet, welcher diametral magnetisiert ist, befestigt. Über diesen kann der absolute Winkel des Rades gemessen werden. Genauere Infos dazu sind im Kapitel Elektronik zu finden. Die vordere Abstützung des Fahrzeuges erfolgt über eine Kugelrolle, damit diese eine so geringe Auswirkung auf die Steuerung hat wie nur möglich.

In der Montage wurden zuerst die beiden Getriebe zusammengesetzt. Einerseits ist es der aufwändigste Teil des Fahrzeugs und anderseits treten hier die grössten Komplikationen auf. Das Getriebe sollte direkt auf die Grundplatte montiert werden so dass gerade alles zusammen passte. Die Schwierigkeit bestand darin, die 2 gelagerten Wellen genau ausgerichtet zu montieren. Damit die Wellen besser positioniert werden konnten wurden die Lagerbohrungen noch um einen Zehntelmillimeter aufgebohrt. Nicht gerade von Vorteil war, dass die Lagerungen der Welle beidseitig eingeklebt werden mussten. Dazu kam, dass der Leim fast eine Stunde brauchte bis er getrocknet war. Während dieser Zeit konnte nicht weiter montiert werden. Ein weiteres Problem war die Genauigkeit der Stirnräder. Einige Zähne waren etwas grösser als die anderen, was dazu führte, dass das Getriebe nicht rund lief. Dies konnte man jedoch mit einem Messer sehr gut zurechtschneiden.

9.3. Greif- und Hebemechanismus

Als Greifer wurde eine einfache Zange mit einem Modellbauservo als Aktor konstruiert. Ein zweites Servo steuert den Hebemechanismus. Die Hebelverhältnisse der Zangenkonstruktion wurden so gewählt, dass mit dem verfügbaren Drehwinkel von etwa 120° den nötigen Öffnungswinkel der Zangenarme erreichen lässt. Modellbauservos haben den Vorteil dass sie günstig sind, genügend Drehmoment haben, schnell genug und einfach anzusteuern sind. Die gesamte Zangenkonstruktion wurde aus ABS im 3D-Printer hergestellt.

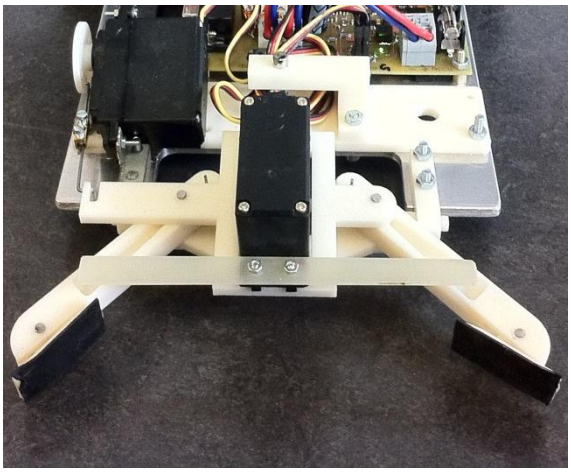


Abbildung 51: Ansicht von Oben

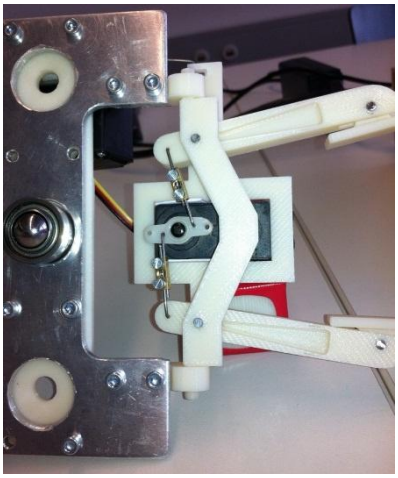


Abbildung 52: Ansicht von Unten

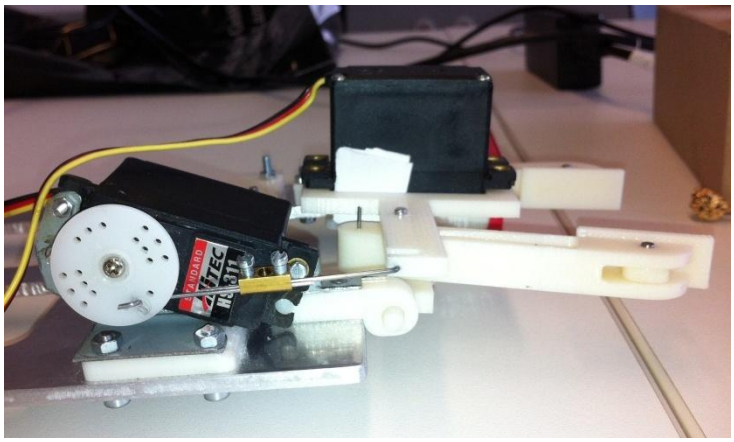


Abbildung 53: Seitenansicht

Eine kurze FEM-Simulation hat gezeigt dass die Belastungen des Materials sehr gering sind:

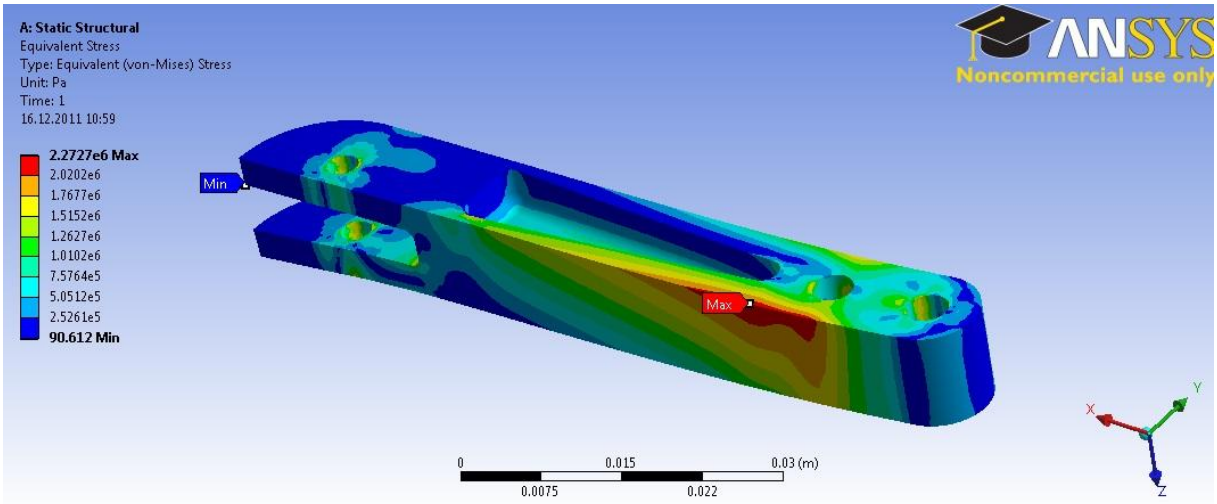
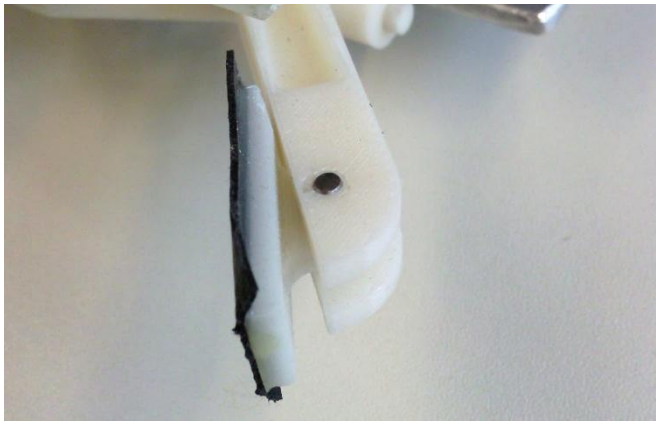


Abbildung 54: FEM-Simulation



Um eine gleichmässige Auflagefläche auf dem Würfel zu erreichen, ist die Greiffläche drehbar gelagert.

Abbildung 55: Greifarm

Bei den ersten Testläufen wurde festgestellt, dass die Greifarme für eine sichere Ergreifung des Würfels länger sein sollten. Die Greifarme wurden darum vorne verlängert um den Würfel besser einfangen zu können.

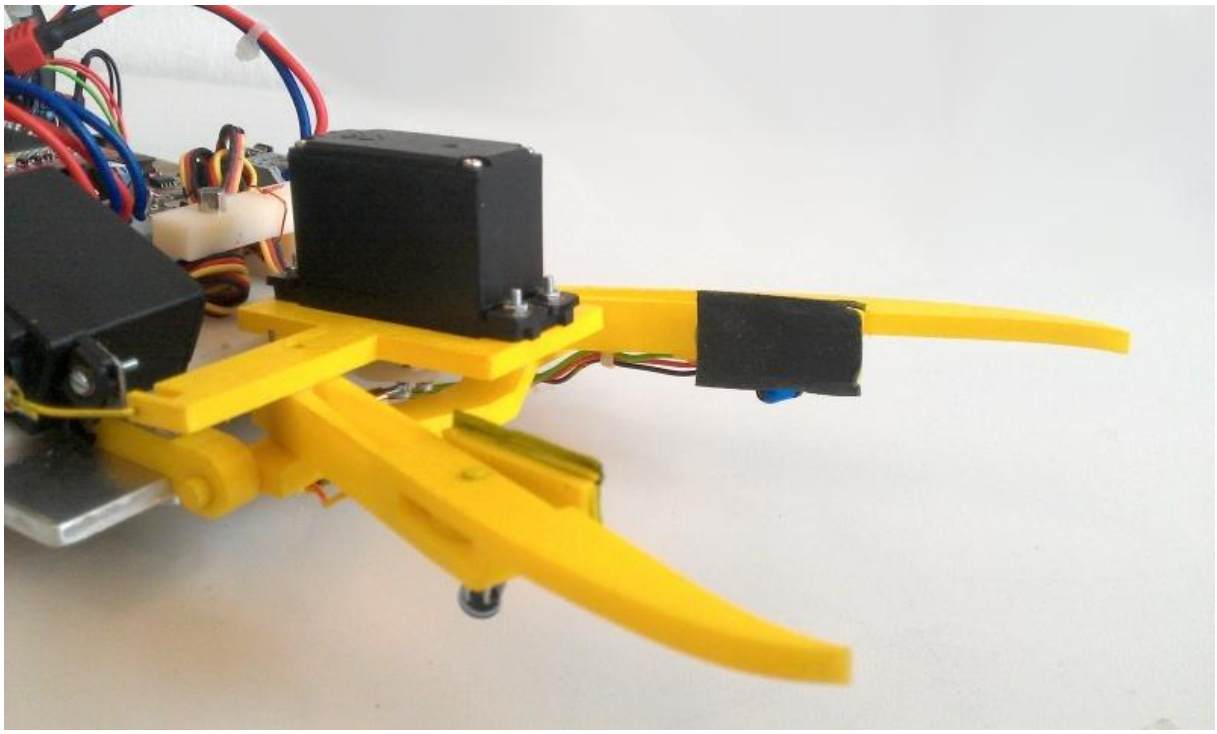


Abbildung 56: Verlängerter und gefärbter Greifer

Um den Würfel sicher greifen zu können, muss der Roboter den Würfel innerhalb des Greifers treffen und ihn ein Stück vor sich her schieben, damit der Würfel in die Mitte des Greifers rutscht. Erst dann kann der Greifer geschlossen und der Würfel gehoben werden.

9.4. Kinect- und Verschalungsbefestigung

Damit die Kinect am Fahrzeug befestigt werden konnte, mussten an der Kamera noch einige Umbauten getätigt werden. Das Problem war, dass sich an der Kinect keine Möglichkeit befand, an der sie befestigt werden konnte. Aus diesem Grund wurde in erster Linie der Fuss der Kamera entfernt um unter der Verschalung eine Möglichkeit zur Befestigung zu finden.

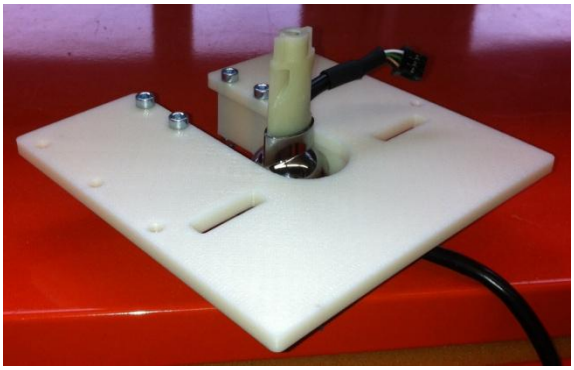


Abbildung 57: Fuss der Kinectkamera

Später jedoch wurde der Wunsch immer grösser, dass die Kinect in der Höhe gekürzt werden soll. Die Kamera funktioniert am besten, je näher sie dem Boden ist. Zudem soll das Fahrzeug damit auch kompakter erscheinen, was vom Design her gewünscht wurde. Aus diesem Grund wurde der gesamte Fuss der Kamera entfernt so dass nur noch das Verbindungsstück zwischen Fuss und Kamera vorhanden war.

An dieses Verbindungsstück wurde dann seitlich je eine Platte montiert. Eine M3 Schraube verbindet die beiden Platten inklusive der Kamera und hält diese somit fest. Die beiden Platten werden dann mit der oberen Querverbindung verschraubt und somit am Fahrzeug befestigt. Der Vorteil dieser Befestigung ist, dass der Winkel, zwischen Kamera und Boden eingestellt werden kann. Später wurden zusätzlich noch 2 Schrauben von unten montiert. Über diese kann jetzt die Neigung der Kamera eingestellt werden, denn ein konstanter Winkel ist die Voraussetzung für konstantes, erfolgreiches Abschliessen des Parcours.

Abbildung 58: Kinectbefestigung von unten

9.5. Optimierungsvorschläge für Serienmodell

Für ein Serienmodell müssen einige Bedingungen erfüllt werden damit das Preis-Leistungsverhältnis möglichst optimiert werden kann. Dabei soll das Produkt möglichst wenig und möglichst einfache Teile enthalten. Dies reduziert die Herstell- und Lagerkosten. Zudem sollte auch der Zeitaufwand für die Montage minimiert werden.

Was die Anzahl und die Einfachheit der Teile betrifft, so kommt man mit dem Fahrzeug schon sehr nahe ans Optimum. Je nach Stückzahl können jedoch noch einige Teile optimiert werden. Zum Beispiel kann das Fahrgestell aus einem einzigen Kunststoffteil hergestellt werden. Somit wäre die Anzahl der Teile schon wieder reduziert. Ab einer gewissen Stückzahl würde sich ebenso lohnen, einen Motor mit integriertem Getriebe herstellen zu lassen. Dies würde Platz im Fahrzeug schaffen und zusätzlich die Anzahl der Teile senken.

Was sicher noch reduziert werden muss, ist der Montageaufwand. Bei diesem Fahrzeug wurde noch zu viel „gebastelt“. Jedoch würde der Montageaufwand schon wesentlich kleiner wenn die obengenannten Optimierungen durchgeführt werden. Im Falle, dass das Getriebe weiterhin selbstgemacht ist, ist hier bestimmt am meisten Verbesserungspotential. Die einzelnen Bestandteile müssen klarer und einfacher montiert werden können. Dazu muss sicher die Lagerung formschlüssig befestigt werden und nicht mehr geklebt wie bisher.

Was die Kamera- und Karosseriebefestigung betrifft, ist sicher auch noch Optimierungsbedarf vorhanden. Die Kamera sollte dann nur noch in einer Position befestigt werden können, ohne dass noch Einstellungen vorgenommen werden müssen. Je nachdem kann man sich auch überlegen, ob man die Kameraelemente, welche jetzt in der Kinect enthalten sind, im Fahrzeug integrieren kann. Dies würde dann aber wieder einen Mehraufwand in der Konstruktion und somit auch in der Fertigung und der Montage bedeuten. Die Verschalung des Fahrzeugs sollte nur über einfache Steckverbindungen montiert und demontiert werden können.

10. Bedienungsanleitung

10.1. Wichtige Sicherheitsanweisungen

Um eine zuverlässige Funktion des autonomen Schnelltransporters (im nachfolgenden Fahrzeug genannt) zu gewährleisten, beachten Sie bitte folgendes:

- Lesen Sie die Anleitung sorgfältig durch und beachten Sie diese bei der Handhabung des Fahrzeugs.
- Verwenden Sie das Fahrzeug nicht draussen. Staub, Schmutz und Wetter können dem Fahrzeug schaden.
- Falls das Fahrzeug von einem kalten in einen warmen Bereich umgestellt wurde, warten Sie 2 bis 3 Stunden bevor Sie das Fahrzeug benutzen. So kann das Kondenswasser, das sich evtl. im Fahrzeug gebildet hat, verdunsten.
- Benutzen Sie das Fahrzeug nicht, wenn es beschädigt ist.
- Reinigen Sie das Fahrzeug mit einem weichen Tuch. Verwenden Sie keine aggressiven Lösungs- oder Reinigungsmittel.

10.2. Leistungsmerkmale

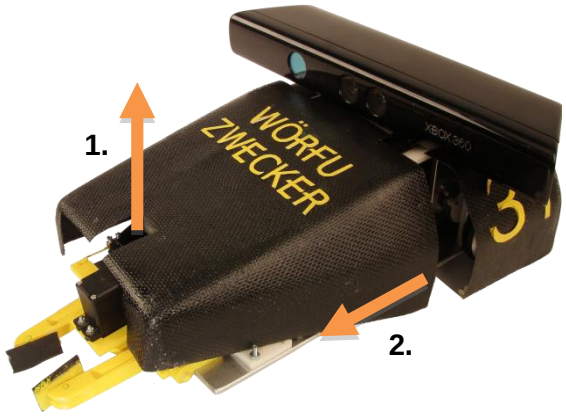
Chassis	Aluminium
Antriebseinheit hinten (links/rechts spiegelbildlich) Abstützung vorne	12V-DC Motoren N2738-051-G5, 11.4W Stirnräder: 65mm Durchmesser, 1:25 Übersetzung Kugelrollenhalter
Drehzahlmessung (links/rechts spiegelbildlich)	Magnetische Encoder mit 10Bit (≈0.2mm Wegstrecke) Auflösung
Greifmechanismus	Greifzange mit Hebemechanismus (je ein Modellbau-Servo), drehbar gelagerte Greiffläche, Material: ABS
Steuerungseinheit	TS-7500 von embeddedARM CPU: 250MHz ARM9 Weiteres: 5K FPGA, 2xUSB, Ethernet, Linux Betriebssystem
Umgebungssensor	Kinect von Microsoft, Tiefenkamera, einstellbare Halterung
Stromversorgung	LiPo-Akkumulator mit 4 Zellen und 2200mAh

10.3. Inbetriebnahme

10.3.1. Demontieren der Abdeckhaube

Die Abdeckhaube ist in einen vorderen und einen hinteren Teil getrennt. Grundsätzlich muss nur die vordere Abdeckhaube entfernt werden. Es ist nicht nötig, irgendwelche Schrauben zu lösen, die (vordere) Abdeckung wird durch Magnete am Chassis festgehalten. Die Haube muss vorsichtig demontiert werden, um sie nicht zu beschädigen.

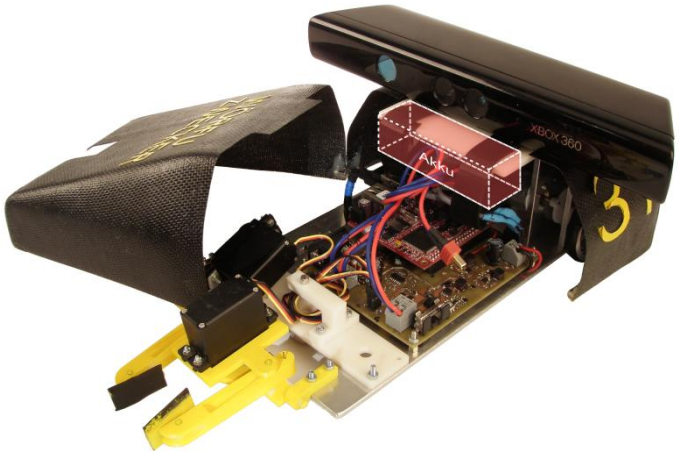
1. Haube bei vorderem Servo nach oben ziehen, bis der vordere Magnet sich löst.
2. Haube schräg nach rechts oder links ziehen, damit sich einer der beiden hinteren Magnete löst.



10.3.2. Einsetzen des Akkus

Nachdem die Abdeckhaube demontiert ist, kann der Akku eingesetzt werden.

1. Den Akku unterhalb der weissen Platte mit den Klettverschlüssen befestigen (siehe Bild).
2. Akkukabel anschliessen.



Der SBC beginnt mit dem Bootvorgang, der ca. 90s dauert und mit zwei kurzen Pieptönen endet.

10.3.3. Laden des Akkus

Laden Sie den Akku nur mit einem dafür vorgesehenen Ladegerät.

10.4. Starten des Programms

10.4.1. Allgemein

Wenn keine neue Konfiguration erforderlich ist, kann das Programm direkt mit dem Starttaster gestartet werden. Die Kinect wird gestartet und das erste Tiefenbild wird aufgenommen. Nach einigen Sekunden ist der Roboter bereit für den Parcours und signalisiert dies durch einen kurzen Piepton. Ein weiterer Druck auf den Startknopf lässt den Roboter sofort losfahren.

Für die Konfiguration des Roboters stehen folgende Schnittstellen zur Verfügung:

- Bluetooth (RFCOMM-Schnittstelle)
- RS232 (Root-Konsole des Kernels)
- Ethernet (SSH)

Jede Variante stellt einen Zugriff auf die Linux-Konsole des SBCs zur Verfügung, über welche die Steuerung konfiguriert und Programme gestartet werden können.

Als Login sind die Benutzer "**root**" und "**eclipse**" eingerichtet, beide mit "eclipse" als Passwort.

Der Roboter hat eine statische IP-Adresse von **192.168.1.150** und eine Bluetooth-MAC von **00:19:0E:0A:63:B7**.

10.4.2. Anforderungen

Für RS232 und SSH kann ein beliebiges Betriebssystem verwendet werden, erforderlich ist nur ein serielles Terminal (bspw. Hyperterminal unter Windows oder minicom unter Linux), bzw. ein SSH-Client (bspw. PuTTY für Windows oder ssh für Linux).

Für Bluetooth-Zugriff wird ein Betriebssystem mit Unterstützung von seriellen Schnittstellen über Bluetooth benötigt. Dafür eignen sich sowohl Linux-Betriebssysteme wie bspw. Ubuntu als auch Windows. Mac OSX wurde nicht getestet.

10.4.3. Setup unter Linux

Falls minicom noch nicht installiert ist, kann es in einem Konsolenfenster mittels

```
sudo apt-get install minicom
```

installiert werden (gilt für Debian-basierte Systeme wie Ubuntu).

Nach der Installation sollte die Default-Konfiguration von minicom angepasst werden, damit diese nicht bei jedem Start neu vorgenommen werden muss. Dazu sollte minicom als Superuser in den Konfigurationsmodus gestartet werden:

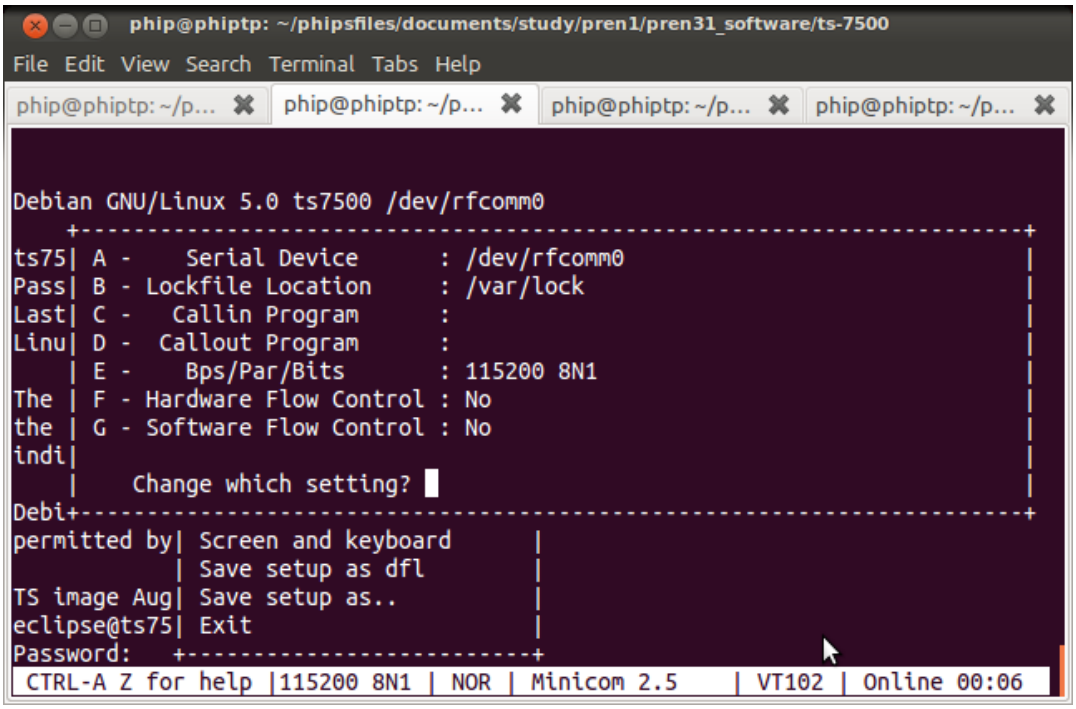
```
sudo minicom -s
```

Anschliessend müssen die Schnittstellenparameter eingestellt werden:

- 115200 Baud
- 8 Datenbits
- 1 Stopbit
- Keine Parität
- Keine Flusssteuerung

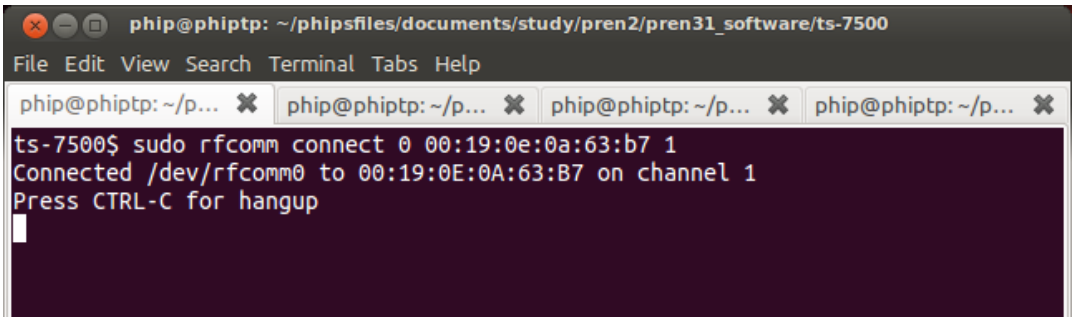
Als "serial port" muss die Gerätedatei der zu verwendenden Schnittstelle eingetragen werden. Diese ist abhängig vom einzusetzenden Übertragungskanal:

- **`/dev/rfcomm0`** für Bluetooth
- **`/dev/ttyS0`** für eine "echte" serielle Schnittstelle (Desktop-PC), COM1
- **`/dev/ttyUSB0`** für einen USB-RS232-Adapter


phip@phiptp: ~/phipsfiles/documents/study/pren1/pren31_software/ts-7500
File Edit View Search Terminal Tabs Help
Debian GNU/Linux 5.0 ts7500 /dev/rfcomm0
A - Serial Device : /dev/rfcomm0
B - Lockfile Location : /var/lock
C - Callin Program :
D - Callout Program :
E - Bps/Par/Bits : 115200 8N1
F - Hardware Flow Control : No
G - Software Flow Control : No
Change which setting?
Screen and keyboard
Save setup as dfl
Save setup as..
Exit
CTRL-A Z for help |115200 8N1 | NOR | Minicom 2.5 | VT102 | Online 00:06

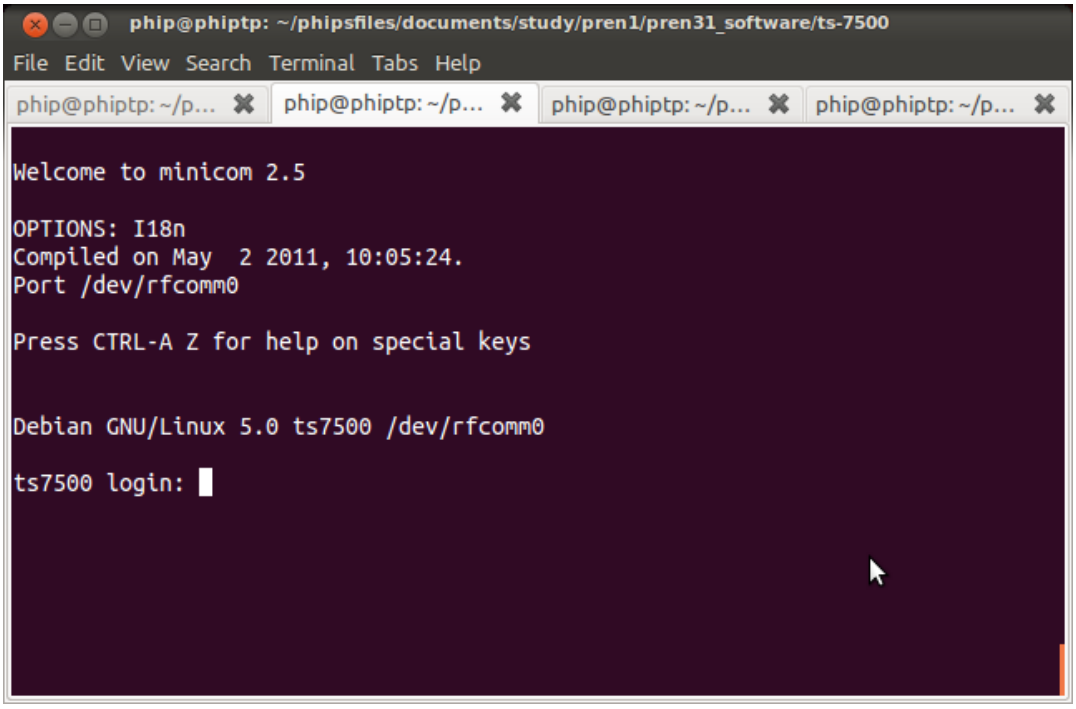
Um die Verbindung zum Roboter herzustellen, muss zuerst eine RFCOMM-Verbindung aufgebaut werden. Dazu sollte ein Konsolenfenster geöffnet und folgender Befehl ausgeführt werden:

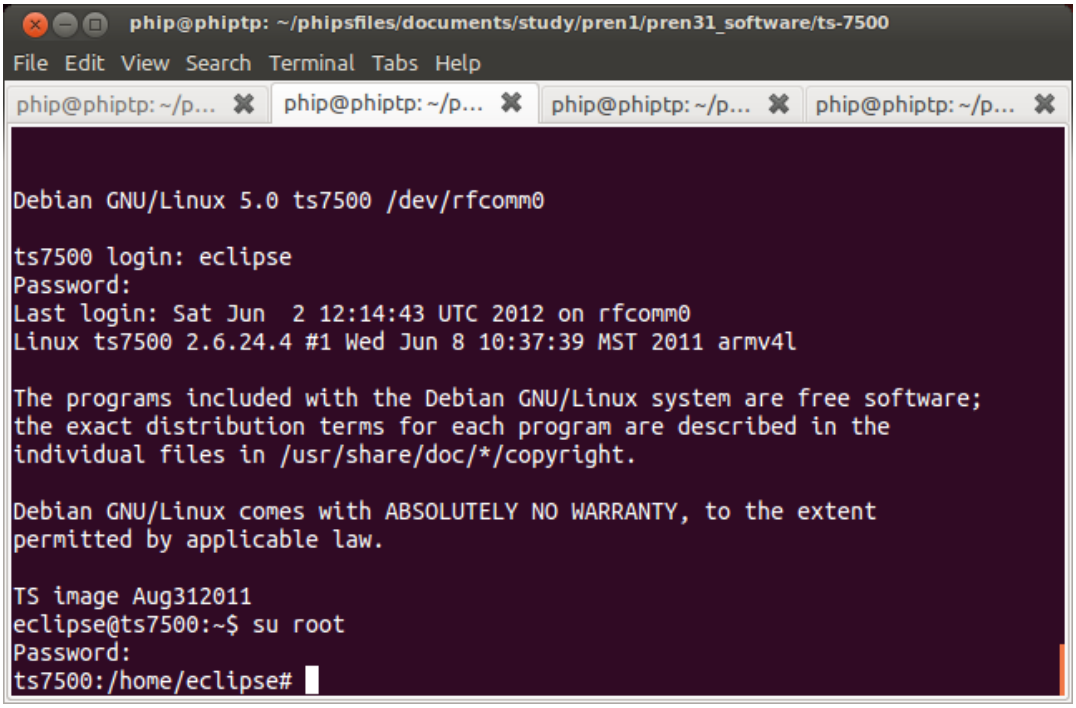
```
sudo rfcomm connect 0 00:19:0e:0a:63:b7 1
```


phip@phiptp: ~/phipsfiles/documents/study/pren2/pren31_software/ts-7500
File Edit View Search Terminal Tabs Help
ts-7500$ sudo rfcomm connect 0 00:19:0e:0a:63:b7 1
Connected /dev/rfcomm0 to 00:19:0E:0A:63:B7 on channel 1
Press CTRL-C for hangup

Anschliessend sollte minicom in einem neuen Konsolenfenster als Superuser gestartet werden:

```
sudo minicom
```




phip@phiptp: ~/phipsfiles/documents/study/pren1/pren31_software/ts-7500
File Edit View Search Terminal Tabs Help
phip@phiptp: ~/p...
phip@phiptp: ~/p...
phip@phiptp: ~/p...
phip@phiptp: ~/p...
Debian GNU/Linux 5.0 ts7500 /dev/rfcomm0
ts7500 login: eclipse
Password:
Last login: Sat Jun 2 12:14:43 UTC 2012 on rfcomm0
Linux ts7500 2.6.24.4 #1 Wed Jun 8 10:37:39 MST 2011 armv4l
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
TS image Aug312011
eclipse@ts7500:~$ su root
Password:
ts7500:/home/eclipse#

Zu diesem Zeitpunkt ist root-Zugriff auf das Linuxsystem verfügbar.

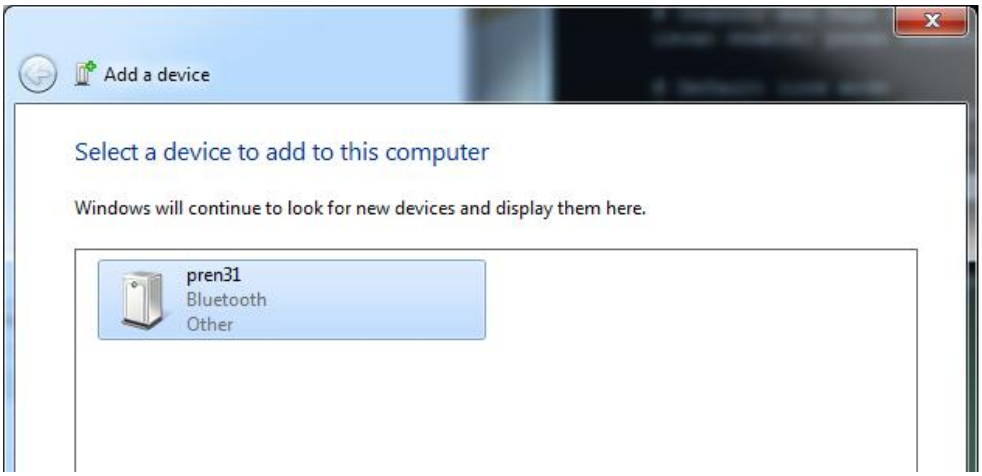
10.4.4. Setup unter Windows

Die folgenden Schritte beziehen sich auf Windows 7 mit englischer Spracheinstellung und PuTTY als Terminalprogramm.

Falls noch nicht vorhanden, sollte PuTTY heruntergeladen werden:
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

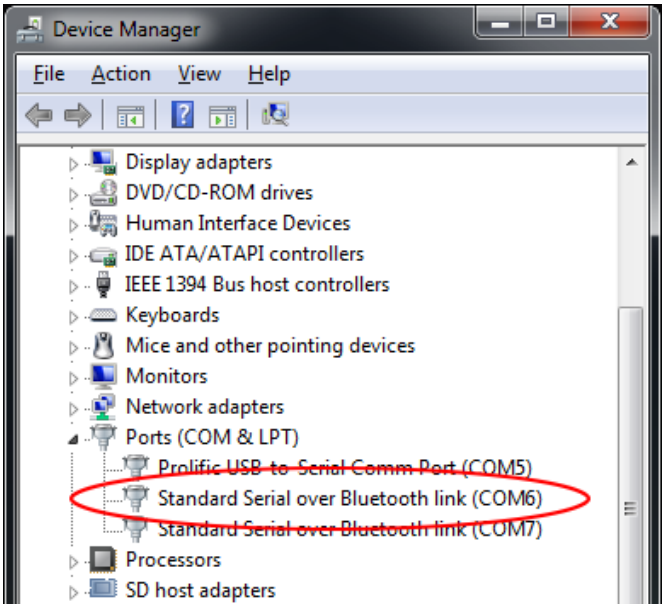
Das Programm erfordert keine Installation, es kann direkt die ausführbare Datei gestartet werden.

Für eine Bluetooth-Verbindung muss vor dem Verbinden im Control Panel (Systemsteuerung) ein neues Bluetooth-Gerät hinzugefügt werden:

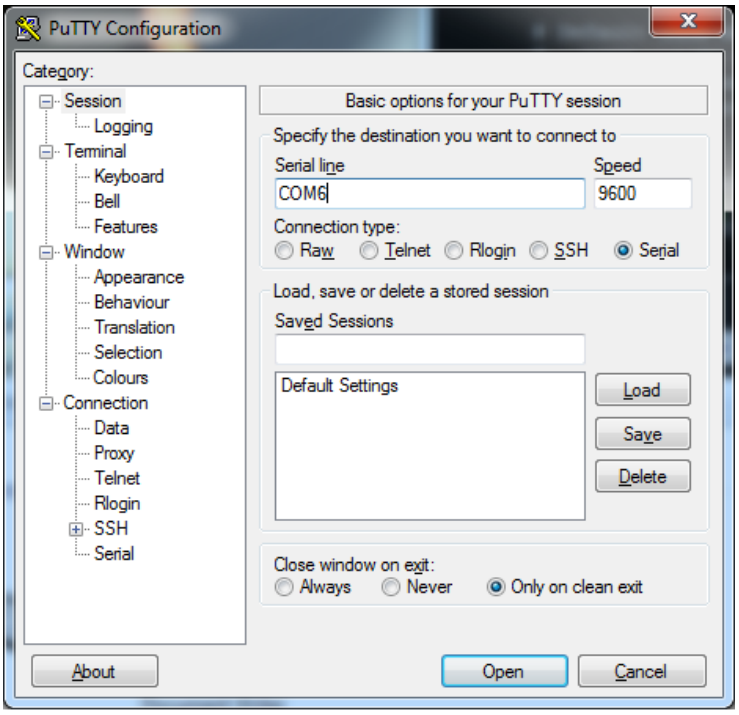

Add a device
Select a device to add to this computer
Windows will continue to look for new devices and display them here.
pren31
Bluetooth
Other

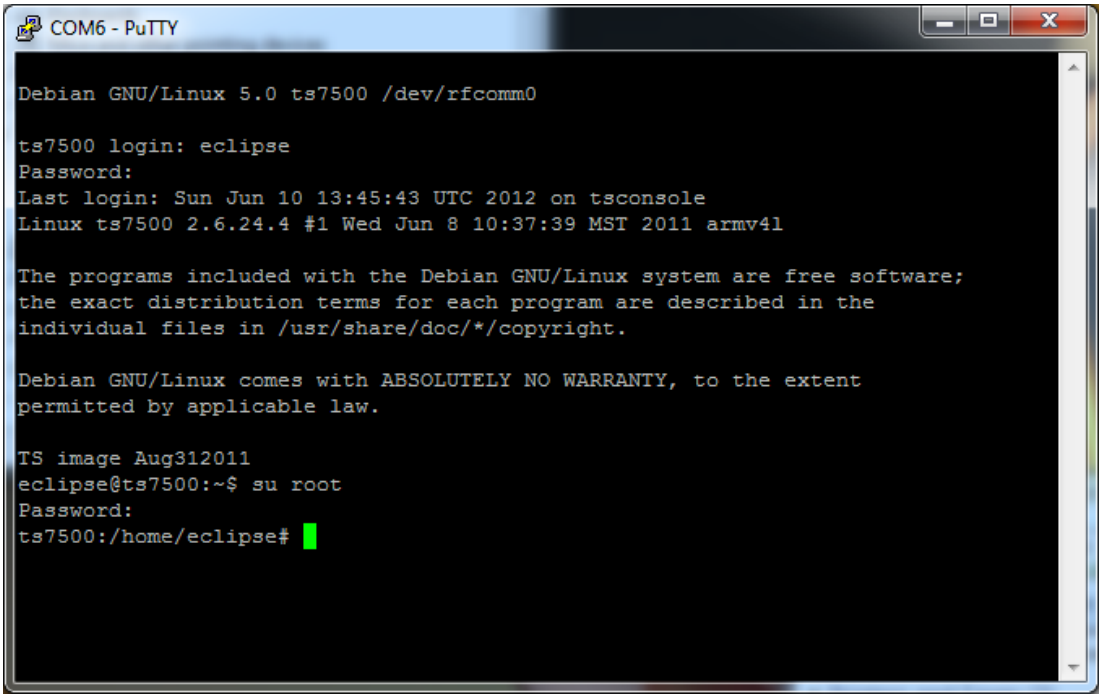
Damit der Roboter gefunden wird muss er eingeschaltet und vollständig gestartet (zwei kurze Piepser) sein und sich in Reichweite (ca. 10m, je weniger desto besser) des Computers befinden.

Der Roboter hat einen fix eingestellten Schlüssel von **2718**, welcher im nächsten Bildschirm eingegeben werden sollte. Alternativ kann auch eine Verbindung ohne Schlüssel hergestellt werden. Wenn die Verbindung korrekt durchgeführt wurde, erscheint im Device Manager (Gerätemanager) mindestens ein "Standard Serial over Bluetooth link":



Die dort angegebenen Schnittstelle (COMx) kann anschliessend in PuTTY wie ein normaler serieller Port verwendet werden. Für Bluetooth ist die Übertragungsrate (Speed) nicht wichtig, für einen regulären seriellen Port muss 115200 Baud eingestellt werden.



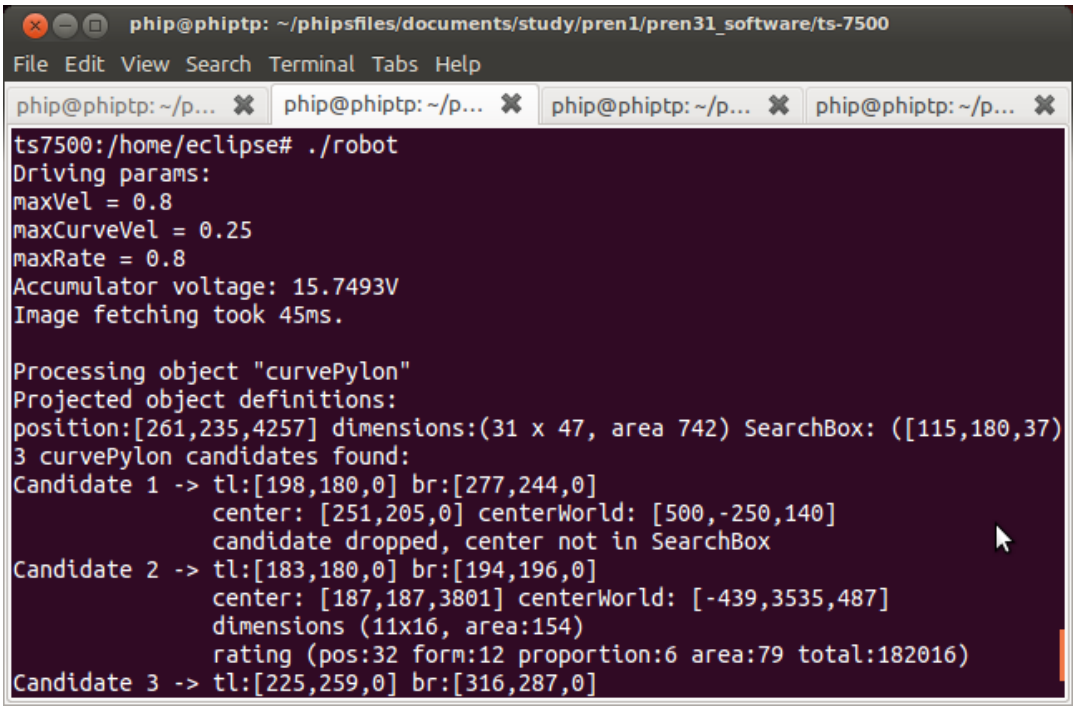

COM6 - PuTTY
Debian GNU/Linux 5.0 ts7500 /dev/rfcomm0
ts7500 login: eclipse
Password:
Last login: Sun Jun 10 13:45:43 UTC 2012 on tsconsole
Linux ts7500 2.6.24.4 #1 Wed Jun 8 10:37:39 MST 2011 armv4l
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
TS image Aug312011
eclipse@ts7500:~$ su root
Password:
ts7500:/home/eclipse#

10.4.5. Programmstart

Nach dem Loginvorgang kann das Roboterprogramm gestartet werden. Dieses befindet sich typischerweise in **`/home/eclipse`** auf dem SBC und trägt den Namen "robot". Es kann mittels

```
./robot
```

gestartet werden.


phip@phiptp: ~/phipsfiles/documents/study/pren1/pren31_software/ts-7500
File Edit View Search Terminal Tabs Help
ts7500:/home/eclipse# ./robot
Driving params:
maxVel = 0.8
maxCurveVel = 0.25
maxRate = 0.8
Accumulator voltage: 15.7493V
Image fetching took 45ms.
Processing object "curvePylon"
Projected object definitions:
position:[261,235,4257] dimensions:(31 x 47, area 742) SearchBox: ([115,180,37)
3 curvePylon candidates found:
Candidate 1 -> tl:[198,180,0] br:[277,244,0]
center: [251,205,0] centerWorld: [500,-250,140]
candidate dropped, center not in SearchBox
Candidate 2 -> tl:[183,180,0] br:[194,196,0]
center: [187,187,3801] centerWorld: [-439,3535,487]
dimensions (11x16, area:154)
rating (pos:32 form:12 proportion:6 area:79 total:182016)
Candidate 3 -> tl:[225,259,0] br:[316,287,0]

Nach dem Start wird der zu umfahrende Pylon im Tiefenbild gesucht. Wird er gefunden, wartet das Programm auf die Bestätigung zum losfahren, ansonsten bricht es ab.

10.5. Problembehandlung

10.5.1. Der Roboter reagiert überhaupt nicht

10.5.1.1. Beschreibung

Es ist kein Zugriff über irgendeine Schnittstelle möglich und es leuchtet keine LED auf der Hauptplatine oder nur die rote direkt neben der Sicherung.

10.5.1.2. Mögliche Ursachen

- Die Sicherung ist defekt
- Die Versorgungsspannung liegt unter 12.5V

10.5.1.3. Behebung

- Kurzschluss entfernen und Sicherung ersetzen (5AT oder 10AT)
- Eine Versorgungsspannung von 13 - 20V verwenden

10.5.2. Der Würfel wird nicht gefunden

10.5.2.1. Beschreibung

Bei der Auswertung des Tiefenbildes wird der Würfel nicht erkannt. Der Roboter fährt direkt ins Ziel ohne zu versuchen, den Würfel aufzunehmen.

10.5.2.2. Mögliche Ursachen

Die Kinect ist nicht am rückseitigen Anschlag und nach vorne geneigt.

10.5.2.3. Behebung

Den Sensorbalken so nach hinten drücken, dass er auf den beiden Schrauben aufliegt.

10.5.3. Regelmässiger Piepton

10.5.3.1. Beschreibung

Der eingebaute Summer piept im Abstand von ca. 5s.

10.5.3.2. Mögliche Ursachen

- Die Akkuspannung ist unter 14.5V gefallen
- Das FPGA wurde nicht korrekt zurückgesetzt.

10.5.3.3. Behebung

- Den Akku laden
- Den Roboter neu starten

10.5.4. Ein Druck auf den Starttaster hat keine Auswirkung

10.5.4.1. Beschreibung

Der Bootvorgang ist abgeschlossen, doch ein Druck auf den Starttaster bewirkt nichts.

10.5.4.2. Mögliche Ursachen

- Die Kinect konnte nicht gestartet werden
- Der `robot-autostartd`-Deamon wurde beendet

10.5.4.3. Behebung

- Startknopf erneut betätigen
- Das Programm über die Konsole starten
- Den Roboter neu booten

11. Diskussion

11.1. Zeitlicher Aufwand

Teilgebiet	Was?	Zuständig	Geschätzt	Effektiv
Mechanik	Konstruktion		110	106
	Detailzeichnungen	am,su	24	3
	Fertigung	am,su	20	4
	Montage	am,su,pb	16	8
	Test	am,su	24	15
	Optimierungen	am,su	20	15
	Unvorhergesehenes	am,su	10	
	Total Mechanik		224	151
Elektronik	Schema	rf	28	32
	Bauteilerfassung	rf	8	10
	Layout	rf	30	40
	Review	rf,pb	4	4
	LP-Fertigung	pb	2	1
	Bestückung	rf,pb	4	4
	Inbetriebnahme	rf,pb	10	6
	Test	rf	8	8
	Optimierungen	rf,pb	10	4
	Total Elektronik		104	109
Software	Softwarekonzepte	jv, pb	25	18
	FPGA	pb	20	28
	Emulator	pb	40	42
	Bilderkennung	jv	23	43
	Antriebssteuerung	jv, pb	10	8
	Orientierung	jv, pb	5	12
	Konfigurationssoftware	jv	25	0
	Tests / Optimierungen	jv, pb	30	50
	Unvorhergesehenes / Allgemein	alle	10	5
	Total Informatik		188	206
Markt	A-Markt bestimmen	mr,rr	6	5
	Wettbewerbsvorteile	mr,rr	8	10
	SWOT-Analyse	mr,rr	4	3
	Produktanforderungen	mr,rr	32	32
	Abweichungen Prototyp-Produkt	mr,rr	16	18
	Verbesserungen	mr,rr	80	62
	Total Markt		146	130
Design	Designstudie	mr,rr	16	18
	Modellierung	am,su,mr,rr	48	44
	Fertigung	am,su,mr,rr	24	20
	Total Design		88	82
Sonstiges	Dokumentation	alle	300	315
Zeitaufwand Total			1050	993

Die Schätzung des Zeitaufwands Ende PREN1 war noch zu ungenau. Während der Zeitplanung anfangs PREN2 wurden die Zeitaufwände erneut geschätzt und den Arbeitspaketen zugeteilt. In der Tabelle sind nur fest zurechenbare Aufwände aufgelistet, d.h. Besprechungen und nicht zuteilbare Zeit sind nicht eingerechnet. In PREN2 gab es wesentlich weniger Diskussionen während den Teamsitzungen als in PREN1. Das liegt daran, dass mehr in den einzelnen Disziplinen gearbeitet wurde. Grobe Mehraufwände traten nicht auf. Der Teilbereich Software benötigte im Vergleich recht viel Aufwand. Da passte es ganz gut, dass der Elektroniker wichtige Teile bei der Software-Entwicklung übernahm und dem Informatiker half. Nicht zu unterschätzen ist der Aufwand für Tests und Optimierungen. Die 80-zu-20-Regel hat sich damit ein weiteres Mal bestätigt.

11.2. Entwicklungskosten

	Beschreibung	*Lieferant*	*Art.-Nr*	*Stück*	*Preis/Stk.*	*Kosten*
Et	ADC, 10bit,SOT23-8,4CH,I2C	Farnell	1607237	1	Fr. 9.05	Fr. 9.05
	Akku			1	Fr. 38.00	Fr. 38.00
	Bluetooth-Dongle			1	Fr. 10.00	Fr. 10.00
	Dist.sens. GP 2D120J000F	Distrelec	631004	1	Fr. 20.30	Fr. 20.30
	Dist.sens. GP 2Y0A02YK0F	Distrelec	631005	1	Fr. 26.40	Fr. 26.40
	Inductor, 0603, 1kOhm	Farnell	1515671	1	Fr. 0.10	Fr. 0.10
	Inductor, 6x6mm, 1uH	Farnell	1782803	1	Fr. 0.61	Fr. 0.61
	Inductor, 6x6mm, 47uH	Distrelec	354417	1	Fr. 1.08	Fr. 1.08
	Kinect Xbox 360			1	Fr. 97.00	Fr. 97.00
	Klemmen (Akku, Motor)			8	Fr. 0.07	Fr. 0.56
	Komparator TL331KDBVR	TI (Samples)		1	Fr. 0.21	Fr. 0.21
	Kondens. 10u/25V	Farnell	1759453	7	Fr. 0.19	Fr. 1.34
	Kondens. 1u/10V, 0603	Farnell	1844204	4	Fr. 0.03	Fr. 0.12
	Kondens. 1u/25V, 1206	Farnell	9227873	7	Fr. 0.04	Fr. 0.25
	Kondens. 470u/25V, 5mm	Farnell	9451200	2	Fr. 0.25	Fr. 0.50
	Kondens. 47u/50V	Farnell	1167997	1	Fr. 0.23	Fr. 0.23
	LM431AIM3, shunt Reg.	Farnell	1215210RL	1	Fr. 1.15	Fr. 1.15
	Magnet, 6x2.5mm, NdFeB	supermagnete	S-06-2.5-DHN	5	Fr. 0.35	Fr. 1.75
	Modularstecker 6/6/6RJ12	Distrelec	129004	2	Fr. 0.50	Fr. 1.00
	OpAmp TLV2374IDR	TI (Samples)		1	Fr. 0.70	Fr. 0.70
	PiezoTransducer, KPEG169	Farnell	1502716	1	Fr. 0.75	Fr. 0.75
	SBC TS7500	Technology Systems		1	Fr. 227.65	Fr. 227.65
	Schirmkabel, 6x0.055mm^2	Distrelec	514829	1	Fr. 3.24	Fr. 3.24
	Schmitt inverter, 74HC14	Farnell	9594566	1	Fr. 0.73	Fr. 0.73
	Schottky 3A/40V	Farnell	1651114	4	Fr. 0.61	Fr. 2.44
	SI4435DY, MOSFET	Farnell	1653655	1	Fr. 1.70	Fr. 1.70
	Sicherungshalter 5x20	Farnell	1162740	1	Fr. 1.20	Fr. 1.20
	Spule 10uH/2.6A	Farnell	1782809	2	Fr. 0.60	Fr. 1.20
	Step-Down-Conv. TPS54140	TI (Samples)		3	Fr. 1.76	Fr. 5.28
	Treiber DRV8412	TI (Samples)		1	Fr. 3.50	Fr. 3.50
	Trimmer SMD 10kOhm	Farnell	1141474	1	Fr. 1.90	Fr. 1.90
	Versand+MWSt1	Distrelec		1	Fr. 12.45	Fr. 12.45
	Versand2	supermagnete		1	Fr. 0.60	Fr. 0.60
	Widerstand 10mOhm	Farnell	1470053	1	Fr. 1.30	Fr. 1.30
	Widerstand 50mOhm	Farnell	1099913	2	Fr. 1.65	Fr. 3.30
Mech	Antriebsmotor			2	Fr. 12.95	Fr. 25.90
	Fahrwerk					Fr. -
	Greifer					Fr. -
	Plexiglas			1	Fr. 21.60	Fr. 21.60
	Servomotor			2	Fr. 7.00	Fr. 14.00
	Stahlwelle			1	Fr. 2.95	Fr. 2.95
	Stirnräder					Fr. -
	Zahnradsortiment			1	Fr. 8.95	Fr. 8.95
Div	Sprühlsonn. Gelb glanz	COOP, Kriens	-	1	Fr. 4.50	Fr. 4.50
	Head Super Comp	Ochsner Sport	5 304417	1	Fr. 9.90	Fr. 9.90
	TOTAL					**Fr. 565.41**

11.3. Lessons Learned

Viele der „Lessons learned„ - Punkte aus PREN1 bestätigten sich auch in PREN2. Ohne ein funktionierendes Team ist jedes Projekt zum Scheitern verurteilt oder hat zumindest mit erheblich mehr Problemen zu kämpfen. In diesem 2. Semester bestätigte sich, dass ein gutes Team aus einem Teamleiter bestehen muss, welcher die Fäden in den Händen hält und den Überblick nicht verliert. Somit wird das Projekt auf der korrekten Bahn gehalten und die Gefahr in eine Sackgasse einzubiegen wird reduziert. Allerdings besteht das Team auch nicht nur aus einer Person. Jede weitere muss ihren Beitrag dazugeben und dafür sorgen, dass aus dem Gegebenen das Beste herausgeholt wird. Jede Person hat ihr Spezialgebiet, auf dem sie ihre Stärken ausspielen kann. Trotzdem ist es von Vorteil, wenn auch grenzübergreifend Wissen ausgetauscht wird, damit jeder überall mitreden kann.

Damit es zu einem positiven Projektabschluss kommen kann, muss zudem von Anfang an, an alles gedacht werden. Keine Details, und scheinen sie auch noch so unwichtig, sollten vernachlässigt oder gar nicht berücksichtigt werden. Das beginnt bei der Hauptfunktion und hört bei der einzelnen Schraube auf. Der Teufel liegt nun mal im Detail. Je besser man die Details kennt, desto besser können auftretende Probleme behoben werden. So hat es sich ausgezahlt, dass der Regler des Fahrzeuges schon von Beginn an sehr genau betrachtet und ausgelegt wurde.

Dies wurde im Bereich der Karosserie zu wenig gemacht. Aus diesem Grund war hier der Bastelfaktor etwas grösser und es mussten öfters verschiedene Varianten probiert werden, was mehr Zeit in Anspruch genommen hat, als wenn dies von Beginn weg klar gewesen wäre. Allgemein sollte in einem Projekt auch das Design von Anfang an genau unter die Lupe genommen werden. Dazu gehören auch vermeintlich vernachlässigbare Kleinigkeiten wie Befestigungsteile. Aber gerade diese sind es, bei denen man sich schlussendlich den Kopf zerbricht, da meist der Platz in der Konstruktion fehlt.

Weitere Knacknüsse sind die viel zu vielen Möglichkeiten die man hat, um ein Problem zu lösen. Hier wird gefordert, dass diese Möglichkeiten auch genau überprüft und mit einander verglichen werden. Zeichnet sich eine Lösung als nicht realisierbar ab, so darf auf keinen Fall noch damit Zeit verschwendet werden. Für solche Fälle sind die Personen gefragt, welche in solchen zeitfressenden Diskussionen das Zepter in die Hand nehmen und dafür sorgen, dass vorwärts gearbeitet wird.

Alles in allem kann man sagen, dass im Team sehr viele positive, wie auch negative Erfahrungen gesammelt werden konnten, welche alle Teammitglieder im beruflichen Leben etwas weiter gebracht haben. Dass das Projekt dann auch noch so positiv abgeschlossen werden konnte gibt einen zusätzlichen Motivationsschub für die nächsten Projekte.

12. Testprotokoll

Beschreibung	OK	NOK	Bemerkungen	Datum + Kürzel
1. Elektronik:				
- Unterspannungs-Abschaltung	X		aus: 12.20V, ein: 12.60V	27.04., pb
- Spannungsregler 5V SBC	X		5.14V	27.04., pb
- Spannungsregler 5.5V Servos	X		5.52V	27.04., pb
- Spannungsregler 12V Kinect	X		12.18V, bei Ubat < 13.7V wird der Ausgang im unbelasteten Fall instabil und fällt auf ca. 11V (Mittelwert) ab	27.04., pb
- Spannungsregler 12V DRV	X		11.82V	27.04., pb
- Motoransteuerung 0% .. ±100%	X		Geringe Überschwinger	27.04., pb
- Servoansteuerung 0% .. 100%	X			27.04., pb
- Abschaltung 12V Kinect	X			27.04., pb
- RS232-Kommunikation	X		115200,8,n,1	27.04., pb
- Piezo-Buzzer 20Hz .. 10kHz	X		Durch Schaltungsmodifikation wesentlich lauter als zuvor	27.04., pb
- Encoder-Auswertung	X			27.04., pb
- Encoderfehler werden erkannt			Nicht geprüft.	14.06., pb
- Tastereingänge	X			27.04., pb
- Spannungsmessung 12V .. 18V	X		±20mV	27.04., pb
- Akkustrommessung 0.5A .. 5A			Nicht geprüft	
- Servostrommessung (2x) 50mA .. 1A	X		±20mA	27.04., pb
2. Unit-Tests der Software				
- Subsystem "blocksim"	X			14.06., pb
- Subsystem "stlloader"	X			14.06., pb
- Subsystem "emulator"	X		Zeitsteuerung ist nicht mehr aktuell	14.06., pb
- Subsystem "kinect"			Keine Unittests vorhanden	14.06., pb
- Subsystem "driver"			Keine Unittests vorhanden	14.06., pb
- Subsystem "utils"	X			14.06., pb
- Gesamtprojekt "robot"	X			14.06., pb
3. Mechanik				
- Getriebe drehen gleichmässig	X			14.06., pb
- Keine losen Schrauben	X			14.06., pb
- Stützrolle dreht leichtgängig	X			14.06., pb
- Antriebsmotor rechts läuft	X			14.06., pb

- Antriebsmotor links läuft	X			14.06., pb
- Servomotor Greifer	X			14.06., pb
- Servomotor Heber	X			14.06., pb
- Würfel klemmen & aufheben	X			14.06., pb
- Kinect hat freie Sicht	X			14.06., pb
- Ausrichtung der Kinect	X		ca. 1° verdreht	14.06., pb
4. Navigation				
- Gerader Pfad	X			14.06., pb
- Kurve 180°, 0.5m Radius, 60%	X		Winkelfehler ca. 6°	14.06., pb
- Beschleunigen 0% .. 80%	X		maxRate 0.8, 1.1s	14.06., pb
- Abbremsen 80% .. 0%	X		maxRate 0.8, 1.1s	14.06., pb
- Parcours bei 20%	X			14.06., pb
- Parcours bei 50%	X			14.06., pb
- Parcours bei 80%	X			14.06., pb
5. Bilderkennung				
- Würfelerkennung mittig, 1m	X			14.06., pb
- Würfelerkennung mittig, 3m	X			14.06., pb
- Würfelerkennung mittig, 0.6m	X			14.06., pb
- Würfelerkennung links, 1m	X			14.06., pb
- Würfelerkennung rechts, 1m	X			14.06., pb
- Pylonerkennung mittig, 1m	X			14.06., pb
- Pylonerkennung mittig, 3m	X			14.06., pb
- Durchschnittliche Erkennungszeit	X		300ms	14.06., pb
6. Gesamtverhalten				
- Verhalten mit Würfel	X		Aufnahmeposition teilweise etwas spät, sonst ok.	14.06., pb
- Verhalten ohne Würfel	X		Fährt direkt ins Ziel	14.06., pb
- Verhalten wenn der Würfel nicht ergriffen wird	X		Keine Melodie im Ziel	14.06., pb
Gemessene schnellste Zeit:	X		10.5s	14.06., pb

13. Literaturverzeichnis

- de.engadget.com; verfügbar unter: http://de.engadget.com/2011/07/04/project-mimicry-neuartige-sandkastenspiele-dank-kinect, Stand 22.03.2012
- de.engadget.com; verfügbar unter: http://de.engadget.com/2011/07/04/project-mimicry-neuartige-sandkastenspiele-dank-kinect, Stand 22.03.2012
- Europäische Sicherheitsnormen für Maschinen, 2007 verfügbar unter: ftp://ftp.cen.eu/cen/Sectors/List/Machinery/ENsonsafetyofmachinery.pdf Stand 02.03.2012
- Europäische Sicherheitsnormen für Maschinen, 2007 verfügbar unter: ftp://ftp.cen.eu/cen/Sectors/List/Machinery/ENsonsafetyofmachinery.pdf Stand 02.03.2012
- Hako Scheuersaugmaschine, verfügbar unter: http://www.hako.com/de_de/Produkte/index.php Stand 09.03.12
- Irobot Angebote, verfügbar unter: www.irobot.com Stand 09.03.12
- Kärcher Scheuersaugmaschine www.kärcher.ch
- Kehrmaschine Wetrok von Abbildung 5: verfügbar unter: http://www.wetrok.ch/cms/cmsAdmin/modules/popup.cfm?picfile=/images/database/katbilder/big/71500_SpeedmaticTwister_V1.0.jpg&width=400&height=400, Stand 23.02.12
- Kobold VR100, 2012 verfügbar unter: http://www.golem.de/1111/88038.html Stand 01.03.12
- Kobold VR100, 2012 verfügbar unter: http://www.golem.de/1111/88038.html Stand 01.03.12
- Lagerhalle verfügbar unter: http://de.123rf.com/photo_2400282_in-einer-lagerhalle.html, Stand 09.03.2012
- Müller-Stevens/Lechner (2011); Müller-Stevens, Günter und Lechner, Christoph: Strategisches Management, 4. Auflage, Stuttgart 2011
- Produktionshalle verfügbar unter: http://www.theaustin.com/Portals/0/CS_fabrik_halle1.jpg Stand: 9.3.2012
- Raum-erkennung mit Route von: http://aktion. kobold.vorwerk.com, Stand 15.03.2012
- Saugroboter-24, 2012 verfügbar unter: http://saugroboter-24.de/ Stand 01.03.12

- Scheuersaugmaschine verfügbar unter, http://www.reinigungsberater.de/scheuersaugmaschine_numatic_ttb_3450-100s,p-49776084.html stand 02.03.2012
- Schraft, R.-D., Hägele, M., Wegener, K. (2004). Service-Roboter-Visionen. München: Hanser Verlag
- Tabelle 1: Marktwirtschaftliche Unternehmen und Beschäftigte nach Grössenklassen, 2008 verfügbar unter: http://www.bfs.admin.ch/bfs/portal/de/index/themen/06/02/blank/key/01/groesse.html, Stand 16.02.12
- Universität Bielefeld: Prof. Dr. Hermann Jahnke http://www.wiwi.uni-bielefeld.de/fileadmin/ctrl/download/Koma_SS10/Kapitel_7.pdf Stand 12.06.2012
- Werkstatt verfügbar unter: http://www.metallbau-boehringer.de/bilder/500_breit/werkstatt_1_500.gif, Stand 15.03.2012